NASA Technical Paper 2375, Pt. I
c.2

October 1984

Transonic Flow Analysis for Rotors

Part 1—Three-Dimensional, Quasi-Steady, Full-Potential Calculation

I-Chung Chang

NASA

NASA
Technical
Paper
2375

1985

Transonic Flow Analysis for Rotors

Part 2—Three-Dimensional, Unsteady, Full-Potential Calculation

I-Chung Chang

Ames Research Center
Moffett Field, California

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

NASA
Technical
Paper
2375

1984

Transonic Flow Analysis for Rotors

Part 1–Three-Dimensional, Quasi-Steady, Full-Potential Calculation

I-Chung Chang

Ames Research Center
Moffett Field, California

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

TABLE OF CONTENTS

SUMMARY

A new computer program is presented for calculating the quasi-steady transonic flow past a helicopter rotor blade in hover as well as in forward flight. The program is based on the full potential equations in a blade-attached frame of reference and is capable of treating a very general class of rotor blade geometries. Computed results show good agreement with available experimental data for both straight- and swept-tip blade geometries.

I. INTRODUCTION

There is an increasing need to develop advanced computational tools for helicopter rotor aerodynamics research. A major thrust to meet this need is to develop a reliable and efficient computer code to predict the transonic flow field over a helicopter rotor blade. Several investigators, using small disturbance theory, have developed computer codes for calculating such flows. Among them, Caradonna and Isom (ref. 1) calculated the flow past a nonlifting, hovering rotor blade. Grant (ref. 2) considered the quasi-steady flow over a nonlifting rotor blade in forward flight. Caradonna (ref. 3) extended his calculation to the unsteady flow past a nonlifting rotor blade in forward flight with simple blade geometry. Finally, Chattot (ref. 4) extended Caradonna's unsteady, small disturbance code to arbitrary blade geometry.

Arieli and Tauber (ref. 5) were first to publish a code ROT22 based on full potential theory for the quasi-steady flow over a rotor blade. Their approach was to modify Jameson and Caughey's widely used fixed-wing computer code FLO22 (ref. 6) to the case of a rotating blade. This method solves the full potential equations in a sheared parabolic coordinate system so that it is possible to treat the blade geometry exactly. In early comparisons with ONERA data, it was found that pressure distributions on the blade were not well-predicted, particularly in the vicinity of swept tips (ref. 7). In the process of verifying a correction to the ROT22 code, it was expedient to develop a new computer code. This new code was denoted TFAR1 (Transonic Flow Analysis for Rotors) and was found to be a useful tool in its own right. It is presented here as an additional method for analyzing the full potential, quasi-steady flow on a rotor blade of arbitrary geometry.

This new code (TFAR1) contains several new features: (1) a new formulation of the problem, (2) the capability of treating cranked blades, (3) the capability of predicting flow over the blade at any azimuthal angle, (4) the option to restrict calculations to the flow over the tip portion of the blade for computational efficiency, and (5) the option to obtain better resolution by clustering grid points at any selected spanwise station.

The computer results obtained from this new code were compared with ONERA test data for straight- and swept-tip blades. The computed results are presented to show that (1) the transonic phenomena that take place on the tip of a rotor blade are

basically three-dimensional and unsteady; (2) the quasi-steady theory predicts good pressure distributions for a rotor blade in flow which is either entirely subsonic or subcritical with weak shocks; and (3) the quasi-steady theory is useful for design work because it gives good pressure distributions for a straight-tip rotor blade near the 90° azimuth where the flow has moderate shock waves.

This report is Part I of a series of planned publications under the same general title, "Transonic Flow Analysis for Rotors."

II. FLOW EQUATIONS

The exact flow field around a helicopter rotor blade in forward flight is generally acknowledged to be a very complex, unsteady, three-dimensional problem. A complete numerical simulation is beyond the state of the art. The flow in the present case is assumed to be inviscid and isentropic. Therefore, a velocity potential, ϕ, exists for the flow described in a frame of reference which is at rest relative to the undisturbed air. In this inertial frame, the complete equation for the velocity potential is

$$\phi_{tt} + [(\vec{\nabla}\phi)^2]_t + \vec{\nabla}\phi \cdot \vec{\nabla}\left[\frac{1}{2}(\vec{\nabla}\phi)^2\right] = a^2\nabla^2\phi \tag{1}$$

where a is the local speed of sound. Bernoulli's equation, relating a and ϕ, is

$$\phi_t + \frac{1}{2}(\vec{\nabla}\phi)^2 + \frac{a^2}{\gamma - 1} = \frac{a_\infty^2}{\gamma - 1} \tag{2}$$

where a_∞ is the sound speed in the undisturbed air, and γ is the specific heat ratio which is equal to 1.4 for air.

If the blade geometry and location are described by $S(\vec{r};t) = 0$, where $\vec{r}$ is the position vector in the inertial frame, then the boundary condition at the blade surface is

$$S_t + \vec{\nabla}\phi \cdot \vec{\nabla}S = 0 \tag{3}$$

For further analysis it is more convenient to implement this surface boundary condition in a moving frame of reference in which the blade location is fixed (fig. 1). Let primed variables refer to the inertial frame, F', and unprimed variables refer to the blade-attached moving frame, F. Suppose that at time, t, the two frames are coincident and that F is moving relative to F' with a linear velocity, $\vec{U}$, and an angular velocity, $\vec{\Omega}$. Then, at time, t, the position vector, $\vec{r}$, of a particular fluid particle is the same for both frames. If a point, P, is rigidly attached in F', it is observed in F to move with velocity $\vec{V} = -(\vec{U} + \vec{\Omega} \times \vec{r})$. Thus, the velocity of fluid particle at P in F is $\vec{q} = \vec{\nabla}\phi + \vec{V}$. The rate of change of ϕ at P is measured by an observer in F as

$$\phi_{t'} = \phi_t + \vec{V} \cdot \vec{\nabla}\phi \tag{4}$$

The potential equation in the moving frame, F, is given by

$$\phi_{tt} + 2\vec{V} \cdot \vec{\nabla}\phi_t + (\vec{V} \cdot \vec{\nabla})(\vec{V} \cdot \vec{\nabla}\phi) + \vec{\nabla}\phi \cdot \vec{V}_t + 2\vec{\nabla}\phi \cdot \nabla\phi_t$$

$$+ 2[(\vec{V} \cdot \vec{\nabla}\phi)\vec{\nabla}\phi] \cdot \vec{\nabla}\phi + \nabla\phi \cdot \nabla\left[\frac{1}{2}(\nabla\phi)^2\right] = a^2\nabla^2\phi \qquad (5)$$

and Bernoulli's equation is

$$\phi_t + \vec{V} \cdot \vec{\nabla}\phi + \frac{1}{2}(\vec{\nabla}\phi)^2 + \frac{a^2}{\gamma - 1} = \frac{a_\infty^2}{\gamma - 1} \qquad (6)$$

Let the moving frame, F, be described in a Cartesian coordinates system in which x, y, and z represent the chordwise, vertical, and spanwise directions of the blade, and whose origin is at the center of rotation. In the inertial frame, let the advance velocity, $\vec{U}$, lie in the (x',z') plane, and let it form the inclination angle, α_0, with the negative x'-axis direction, and also let the angular velocity, $\vec{\Omega}$, be in the positive y'-axis direction. The velocity, $\vec{V}$, caused by the motion of the frame, F, has components of

$$V_1 = \Omega z + U \cos \alpha_0 \sin \psi$$

$$V_2 = U \sin \alpha_0$$

and

$$V_3 = -\Omega x + U \cos \alpha_0 \cos \psi$$

where ψ is the azimuthal angle of the blade ($\psi = 180°$ for forward flight direction in the inertial frame).

The potential equation in Cartesian coordinates is

$$\phi_{tt} + 2q_1\phi_{xt} + 2q_2\phi_{yt} + 2q_3\phi_{zt}$$

$$= (a^2 - q_1^2)\phi_{xx} + (a^2 - q_2^2)\phi_{yy} + (a^2 - q_3^2)\phi_{zz} - 2q_1q_2\phi_{xy} - 2q_1q_3\phi_{xz} - 2q_2q_3\phi_{yz}$$

$$+ (\Omega^2 x - 2\Omega U \cos \alpha_0 \cos \psi)\phi_x + (\Omega^2 z + 2\Omega U \cos \alpha_0 \sin \psi)\phi_z \qquad (7)$$

where q_1, q_2, and q_3 are the velocity components of local fluid particle in the moving frame and are specified as

$$q_1 = \phi_x + V_1$$

$$q_2 = \phi_y + V_2$$

and

$$q_3 = \phi_z + V_3$$

This equation is similar to the one presented in reference 5 with the exception of the last two terms.

Bernoulli's equation in Cartesian coordinates is

$$\phi_t + V_1\phi_x + V_2\phi_y + V_3\phi_z + \frac{1}{2}(\phi_x^2 + \phi_y^2 + \phi_z^2) + \frac{a^2}{\gamma - 1} = \frac{a_\infty^2}{\gamma - 1} \tag{8}$$

The present study is focused on the three-dimensional effect. For steady calculation, all time-dependent terms in equations (7) and (8) are dropped. Note that the effect of rotation is still included since it is always present in the transformation mapping. This steady case is called quasi-steady in the helicopter literature. Several boundary conditions are necessary to complete the boundary value problem.

In the near field, the flow tangency to the blade is described by the expression

$$\vec{q} \cdot \vec{n} = 0$$

where $\vec{n}$ is the normal unit vector to the blade surface. The wake that is shed from the trailing edge is assumed to be a vortex sheet which is a smooth continuation of the trailing edge. Across this vortex sheet, the pressure is assumed to be continuous. The jump in potential determined at the trailing edge of each spanwise profile is then assumed to propagate to infinity instantaneously. At the far field, the boundary condition can be formulated as a Dirichlet condition where the potential vanishes.

III. MESH SYSTEM

It is much simpler to include boundary condition in a finite difference calculation if the boundary surface is conformal with the coordinate surface. A parabolic sheared mesh system that is employed in the present analysis is similar to one that is used in previous analyses with fixed wings (ref. 6). The mesh system is generated by a series of transformations from the physical space to the computational domain (fig. 2).

First, the shearing transformation

$$\left.\begin{aligned} \bar{x} &= x - x_s(z) \\ \bar{y} &= y - y_s(z) \\ \bar{z} &= z \end{aligned}\right\} \tag{9}$$

shears out the blade sweep and dihedral. Here, the point $x_s(z)$, $y_s(z)$ is the center of the circle passing through three points near the leading edge of the profile at each spanwise station. Second, the scaling transformation

$$\left.\begin{aligned} \tilde{x} &= \bar{x}/SCAL \\ \tilde{y} &= \bar{y}/SCAL \\ \tilde{z} &= \bar{z}/SCALZ \end{aligned}\right\} \tag{10}$$

accounts for the scaling between the physical space and the computational domain. Third, the square root transformation

$$\left.\begin{aligned} (X_1 + iY_1)^2 &= 2(\tilde{x} + i\tilde{y}) \\ Z_1 &= \tilde{z} \end{aligned}\right\} \tag{11}$$

maps the entire blade surface to a shallow bump $Y_1 = S(X_1, Z_1)$ near the plane $Y_1 = 0$. Fourth, the second shearing transformation

$$\left.\begin{aligned} X &= X_1 \\ Y &= Y_1 - S(X_1, Z_1) \\ Z &= Z_1 \end{aligned}\right\} \tag{12}$$

reduces the blade surface to a portion of the plane $Y = 0$. Finally, the stretching transformations are introduced to render the computational domain finite. For example,

$$Y = \frac{b\bar{Y}}{(1 - \bar{Y}^2)^a} \qquad b > 0 \;, \; 0 < a < 1$$

is used to map the planes $Y = \pm\infty$ to $\bar{Y} = \pm 1$. Similar transformations are used outboard of the blade tips in the Z-direction and downstream of the trailing edge in the X-direction. At the blade trailing edge, the branch cut in each spanwise plane is continued smoothly downstream. This cut will be taken as the location of the vortex sheet across which the wake condition is applied.

The transformations (9)-(12) applied to the steady form of equation (7) yield an equation of the form

$$A\phi_{XX} + B\phi_{YY} + C\phi_{ZZ} + 2D\phi_{XY} + 2E\phi_{XZ} + 2F\phi_{YZ} + R_1\phi_X + R_2\phi_Y + R_3\phi_Z = 0 \tag{13}$$

If we introduce the following notation,

$$\begin{aligned} \sigma &= X_{1_{\tilde{x}}} \\ \mu &= X_{1_{\tilde{y}}} \\ \xi &= -(x_s'\sigma + y_s'\mu) \\ \eta &= x_s'\mu - y_s'\sigma \\ \alpha &= -(S_X\sigma + \mu) \\ \beta &= \sigma - S_X\mu \\ \zeta &= \mathrm{SCAL/SCALZ} \\ \gamma &= \eta - \xi S_X - \zeta S_Z \end{aligned}$$

$$\chi = x_s' X_{1_{\tilde{x}\tilde{x}}} + y_s' X_{1_{\tilde{x}\tilde{y}}}$$

$$\Psi = x_s' X_{1_{\tilde{x}\tilde{y}}} - y_s' X_{1_{\tilde{x}\tilde{y}}}$$

$$\Lambda = S_X X_{1_{\tilde{x}\tilde{y}}} + X_{1_{\tilde{x}\tilde{y}}}$$

$$\Sigma = S_X X_{1_{\tilde{x}\tilde{y}}} - X_{1_{\tilde{x}\tilde{y}}}$$

$$\bar{U} = q_1\sigma + q_2\mu + q_3\xi$$

$$\bar{V} = q_1\alpha + q_2\beta + q_3\gamma$$

$$\bar{W} = q_3$$

$$\pi = y_s'\Psi + x_s'\chi$$

$$\lambda = y_s'\chi - x_s'\Psi$$

$$\theta = S_X\chi + \Psi$$

$$\kappa = S_X\Psi - \chi$$

$$\varepsilon = x_s''\sigma + y_s''\mu$$

$$\delta = x_s''\alpha + y_s''\beta$$

$$L = \pi - \varepsilon SCAL$$

$$M = (\Omega^2 x - 2\Omega U \cos\alpha_0 \cos\psi) \cdot SCAL$$

$$N = (\Omega^2 z + 2\Omega U \cos\alpha_0 \sin\psi) \cdot SCAL$$

$$P = \lambda - S_X\pi$$

and

$$Q = R - \delta \cdot SCAL$$

Then, the coefficients of equation (13) can be written as

$$A = \bar{U}^2 - a^2(\sigma^2 + \mu^2 + \xi^2)$$

$$B = \bar{V}^2 - a^2(\alpha^2 + \beta^2 + \gamma^2)$$

$$C = \zeta^2(\bar{W}^2 - a^2)$$

$$D = \bar{U}$$

$$E = \bar{V}$$

$$F = \bar{W}\zeta$$

$$R_1 = \sigma M + \xi N + (q_1^2 - q_2^2)X_{1_{\tilde{x}\tilde{x}}} + (q_3^2 - a^2)L + 2q_1q_2X_{1_{\tilde{x}\tilde{y}}} - 2q_1q_3\chi - 2q_2q_3\Psi$$

$$R_2 = \alpha M + \gamma N + a^2[(\sigma^2 + \mu^2 + \xi^2)S_{XX} + \zeta^2 S_{ZZ} + 2\zeta\xi S_{XZ} + Q] - (q_1^2 - q_2^2)\Lambda$$
$$- \bar{U}^2 S_{XX} - q_3^2\zeta^2 S_{ZZ} - 2\bar{U}q_3\zeta S_{XZ} - q_3^2 Q - 2q_1q_2 \Sigma + 2q_1q_3\theta + 2q_2q_3\kappa$$

and

$$R_3 = \zeta N$$

At the blade surface, the tangential flow condition is simply $\bar{V} = 0$ in the X,Y,Z coordinates. At the far field, the Dirichlet condition $\phi = 0$ is imposed in the present study. For points on the continuation of the singular line outboard of the blade tips, where the Jacobian vanishes, the potential equation reduces to the Laplace equation

$$\phi_{XX} + \phi_{YY} = 0$$

IV. FINITE DIFFERENCE APPROXIMATION

The potential equation can be rearranged in the canonical form

$$(a^2 - q^2)\phi_{ss} + a^2(\nabla^2\phi - \phi_{ss}) + \text{first-order terms} = 0 \tag{14}$$

where q is the magnitude of the velocity, $\vec{q}$, and s is the local flow direction. This equation is elliptic for subsonic flow ($q < a$) and is hyperbolic for supersonic flow ($q > a$). At subsonic points, central differences are employed to approximate all derivatives. At supersonic points, upwind differences are applied to ϕ_{SS} of the first term of equation (14) whereas central differences are used to approximate the rest of the terms of equation (14). In the X,Y,Z system,

$$q^2\phi_{ss} = \bar{U}^2\phi_{XX} + \bar{V}^2\phi_{YY} + \bar{W}^2\phi_{ZZ} + 2\bar{U}\bar{V}\phi_{XY} + 2\bar{U}\bar{W}\phi_{XZ} + 2\bar{V}\bar{W}\phi_{YZ} \tag{15}$$

It is essential for rotor flow calculation to apply upwind differences to all the derivatives in expression (15) in all three directions according to the sign of U, V, and W.

V. SOLUTION ALGORITHM

A generalized line relaxation scheme is used to solve the finite difference approximations of the flow equations in X,Y,Z system. A typical central difference formula for ϕ_{XX} is

$$\phi_{XX} = \frac{\phi^{n+1}_{i-1jk} - (2/\omega)\phi^{n+1}_{ijk} - 2(1 - 1/\omega)\phi^{n}_{ijk} + \phi^{n}_{i+1jk}}{\Delta X^2}$$

where the superscripts denote the iteration level and ω is the relaxation factor.

Similarly,

$$\phi_{XY} = \frac{\phi^{n}_{i+1,j+1,k} - \phi^{n}_{i+1,j-1,k} - \phi^{n+1}_{i-1,j+1,k} + \phi^{n+1}_{i-1,j-1,k}}{4\ \Delta X\ \Delta Y}$$

At supersonic points, for positive U and V, typical upwind differences are

$$\phi_{XX} = \frac{2\phi^{n+1}_{ijk} - \phi^{n}_{ijk} - 2\phi^{n+1}_{i-1,jk} + \phi^{n}_{i-2,jk}}{\Delta X^2}$$

and

$$\phi_{XY} = \frac{\phi^{n+1}_{ijk} - \phi^{n+1}_{i-1,jk} - \phi^{n+1}_{ij-1,k} + \phi^{n+1}_{i-1,j-1,k}}{\Delta X\ \Delta Y}$$

The relaxation process can be regarded as an approximation to some artificial time-dependent equation if we regard each iteration as representing an advance Δt, in artificial time coordinate (ref. 8). An additional term of the form

$$\beta\ \Delta t\phi_{st} = \beta\ \Delta t[\bar{U}\phi_{Xt} + \bar{V}\phi_{Yt} + \bar{W}\phi_{Zt}]\ , \qquad \beta > 0$$

has been added to this artificial time-dependent equation to speed up the convergence rate of the scheme. The upwind differences are used to approximate the spatial derivatives of this term. This implicitly introduces a convective viscosity to the equation and the scheme is further stabilized. The resulting linear system for the unknown ϕ^{n+1}_{ijk} is very large. However, its horizontal lines (j and k constant) are decoupled. Each horizontal line can thus be solved by a tridiagonal matrix solver.

VI. RESULTS AND DISCUSSION

A typical run consists of 100 relaxation sweeps on each of three different grids (a finer grid containing twice as many grid points in each direction of a coarse grid). An initial calculation is performed on a coarse grid containing 32 by 6 by 8 grid points in the X, Y, and Z directions, respectively. The solution is then

interpolated onto a medium grid and is used as a starting guess. The process is repeated again for the fine grid to get the final solution. A typical run for each azimuthal position takes about 40 sec (CPU time) on the NASA Ames Cray 1-S computer.

Comparisons are made with experimental data from two model rotor blades that were tested at ONERA in 1978. The detailed blade geometries, one of which had a swept tip, are described in reference 9. Both blades are tapered and have symmetric blade sections. The swept-tip blade has a 30° leading edge sweep on the outer 15% of the blade (the kink is at $r/R = 0.85$). Their geometries, figures 3(a) and 3(b), are approximated in TFAR1 by the respective geometries, figures 3(c) and 3(d). The trailing edge of the approximate blades is sawtoothed.

The first set of results presented is for the nonlifting straight tip blade at a free-stream Mach number of 0.2406 ($q_\infty = 80.4$ m/sec) and a tip Mach number of 0.5976 because of rotation ($\Omega R = 199.7$ m/sec). The advance ratio is about $\mu = 0.4$. Figure 4 compares the calculated and measured surface pressure distributions at three different span stations, $r/R = 0.85$, 0.9, and 0.95 for azimuthal angles from 0° to 180° at 30° increments. Agreement is good for this case. It is noted that the flows are either entirely subsonic or subcritical with a small supersonic zone. In other words, when the unsteady effect of the flow is small, the code predicts good pressure distributions.

The second set of results that is presented is for the same straight tip blade at a free-stream Mach number of 0.3292 ($q_\infty = 110$ m/sec) and at a tip Mach number of 0.5976 because of rotation ($\Omega R = 199.7$ m/sec). The advance ratio is high ($\mu = 0.55$). Figure 5 compares the calculated and measured surface pressure distributions at the same three span stations ($r/R = 0.85$, 0.9, and 0.95) for azimuthal angles from 0° to 330° at 30° increments. Overall, agreement is fine for the advancing flow side, and is poor for the reverse flow side. The flow fields in the advancing flow side are subsonic with moderate or greater zones of embedded supersonic flow. The code predicts stronger shock waves in the first quadrant and predicts weaker shock waves in the second quadrant when compared with the ONERA data. It should be pointed out that the code predicts good pressure distributions near 90° azimuth in spite of the unsteady effect that is quite strong there. A comparison between TFAR1 and ROT22 results (ref. 5) at span station ($r/R = 0.9$) for the azimuthal angles ($\psi = 60$, 90, and 120) is shown on figures 5(c)-5(e). The differences may be due to the absence of terms in the flow equation as we mentioned (eq. 7).

A similar calculation is performed for the swept blade at a free-stream Mach number of 0.3127 ($q_\infty = 105$ m/sec), a tip Mach number 0.6288 ($\Omega R = 210$ m/sec) caused by blade rotation and a 0° angle of attack. The advance ratio is $\mu = 0.5$. Figure 6 shows computed and experimental surface pressure distributions for this case. The prediction with TFAR1 in the vicinity of the crank is good. TFAR1 accounted for the leading-edge sweep and its effect on the pressure distribution. One of the effects of the sweep-back of the blade tip is to delay the shock formation. This can be seen from the fact that the code TFAR1 predicts good pressure distributions at the 120° azimuth for the 30° swept-tip blade.

VII. CONCLUSIONS

A finite difference code, TFAR1, for predicting quasi-steady transonic flow over a helicopter rotor blade was presented. The code solves the second order full-potential equation in the moving frame and is suitable for modeling the thickness effect of the blade.

Computed results obtained from this new code have been compared with ONERA data for both straight- and swept-tip blades with advance ratios ranging from 0.4 to 0.55. Results showed excellent comparisons between quasi-steady computations and experimental pressure distributions for flow which was entirely subsonic or subsonic with a small supersonic zone. Fair correlation between quasi-steady computational and experimental pressure distributions were obtained for flow with moderate or greater zones of embedded supersonic flow.

It is concluded that (1) quasi-steady theory can predict good pressure distributions for flows without any shock or with weak shocks, (2) quasi-steady theory can still predict good pressure distributions for a straight-tip blade near 90° azimuth for flows having moderate shocks and thus is good for design work, (3) the unsteady effect that takes place on the tip of a rotor blade on the advancing side is basically caused by the transient shock movement, and (4) an unsteady theory is necessary to predict the flow field around a helicopter rotor blade when shocks of moderate strength appear.

APPENDIX A

DESCRIPTION OF THE CODE

The input data deck consists of sequences of pairs of cards. The first card of each pair gives the names of the parameters that appear on the data cards that follow. All data items are read as floating point numbers in a field of 10 columns, and values that represent integer parameters are converted in the program. All the input data is immediately printed as output so that it is easy to check the accuracy of the input.

After the flight condition is read in, the blade geometry is defined by giving blade section profiles at successive spanwise stations from blade root (near the center of revolution) to blade tip. The blade planform and dihedral are determined by specifying the chord, the leading edge coordinates, and twist angle at each sectional profile. After the first airfoil is read in, only the leading edge coordinates, the chord, and the twist angle are given at the new station if this new sectional profile is similar. Otherwise, the input profile should be provided again. The blade sections between two given stations are generated by interpolation. The program prints the coordinates of the unfolded sectional profiles that are produced by the code at the root and at the tip of the blade. They should be inspected to see if they are reasonably smooth.

The program also prints a chart of values of an indicator-IV which shows the characteristics of points in the $Y = 0$ plane. The indicator-IV = 2 indicates a point on the blade, IV = 1 indicates a point on the trailing vortex sheet, IV = 0 indicates a point on the singular line, IV = -1 indicates a point adjacent to the edge of the blade on the vortex sheet, and IV = -2 indicates an ordinary point beyond the blade or vortex sheet.

The program next displays the iteration history. The maximum correction to the velocity potential and the maximum residual of the difference equation together with the i, j, and k location, the relaxation factors, the circulation at the middle blade section, and the number of supersonic points are printed at every cycle.

After a specified maximum number of cycles has been completed, or a convergence criterion has been satisfied, the section lift, drag, and moment coëfficients are printed for each span station and the pressure distribution is printed or displayed in a plot as desired. Finally, the characteristics of the blade are printed which include the coefficients of lift and form drag that are computed by integrating the surface pressure. An estimate of friction drag coefficient may be supplied in the input, and this will be included to produce an estimate of the total drag coefficient. At the end, additional plots are generated if they are desired. These show a view of the blade and the three-dimensional pressure distributions over the upper and lower surfaces, respectively, with the root at the bottom of the picture.

APPENDIX B

GLOSSARY OF INPUT PARAMETERS

TITLE — Title of the case being run. (A format)

Card pair 1:

FNX — The number of mesh intervals in the direction of the chord.

FNY — The number of mesh intervals in the direction normal to the chord and span.

FNZ — The number of mesh intervals in the direction of the span.

Card pair 2:

FIT — The maximum number of iteration cycles which will be computed.

COVO — The desired accuracy.

P10 — The subsonic relaxation factor for the velocity potential. P10 lies between 1 and 2 and should be increased toward 2 with mesh refinement.

P20 — The supersonic relaxation factor for the velocity potential. Recommended value 1.

P30 — The relaxation factor for the circulation. Recommended value 1.

BETAO — The damping factor which controls the amount of convective term. Recommended value 0.1.

FHALF — Determines whether the mesh will be refined. FHALF = 0 terminates the computation after FIT iterations or after convergence. FIT = 0 halves the mesh after FIT iterations or convergence on the coarse mesh. An additional card pair 2 is required for each mesh refinement. The value FHALF = 0 appears on the last mesh refinement card.

Card pair 3:

FSPEED — The forward flight speed (m/sec).

PSI — The azimuthal angle of the blade (deg).

ALPHA — The angle of attack (deg).

TIPWR — The tip speed due to the rotation of the blade (m/sec).

RADIUS — The rotor disk radius (m).

AINF — The speed of sound of undisturbed air in far field (m/sec).

Card pair 4:

CREF The reference chord length.

XREF The reference chordwise ordinate of point about which the sectional airfoil pitching moment coefficient is calculated.

FBLADE Controls the tip portion of blade to be calculated. FBLADE = 1 gives the whole blade.

FCLUST Controls the spanwise mesh point distribution. FCLUST = 0 means uniform grid is used.

CDO The estimated drag due to skin friction. This can be added to the drag calculated by the code to give the total drag.

Card pair 5:

FNC The number of span stations from the blade root to the tip.

SWEEP1 The sweep of the singular line at the blade root (deg).

SWEEP2 The sweep of the singular line at the blade tip (deg).

SWEEP The sweep of the singular line in the far field (deg).

DIHED1 The flap angle of the singular line at the blade root (deg).

DIHED2 The flap angle of the singular line at the blade tip (deg).

DIHED The flap angle of the singular line in the far field (deg).

Card pair 6:

ZS Span location of the section.

XL X coordinate of the leading edge.

YL Y coordinate of the leading edge.

CHORD The local chord value by which the profile coordinates are scaled.

THICK Modified the section thickness. The Y coordinates are multiplied by THICK.

TWIST The angle through which a section is rotated to introduce twist about the quarter-chord point of the section airfoil.

FSEC Indicates whether or not the geometry for a new profile is supplied. FSEC = 0 means the section is obtained by scaling the profile used at the previous span section according to the parameters CHORD, THICK, AND TWIST. No further cards are read for this span station and the next card is the title card for the next span station, if any. FSEC = 1 means the coordinates for a new profile are to be read from the data cards that follow.

Card pair 7:

YSYM — Indicates the type of profile. YSYM = 0 means the data supplied are for a cambered profile. Coordinates are given for the upper and the lower surfaces, each ordered from nose to tail with the leading edge included in both surfaces. YSYM = 1 means the data supplied are for a symmetric profile. A table of coordinates is read in for the upper surface only.

FNU — The number of upper surface coordinates.

FNL — The number of lower surface coordinates. For YSYM = 1, NL = NU.

Card pair 8: (Upper surface coordinates)

X,Y — The coordinates of the upper surface. They appear from leading edge to trailing edge.

Card pair 9: (Lower surface coordinates)

X,Y — The coordinates of the lower surface from leading edge to trailing edge. The leading edge of the upper surface is the same as the leading edge of the lower surface. The trailing edge points are different if the profile has an open tail.

Card pairs 10, 11, . . .:

These card pairs are like card pairs 6, 7, 8, and 9. The number of such card pairs depends on the number of span stations, FNC.

APPENDIX C

LISTING OF TFAR1 PROGRAM

```
*COMDECK BLANK
      COMMON/ /    G(129,18,33),S0(129,33),E0(33),
     .             IV(129,33),ITE1(33),ITE2(33),SMACH(33),
     .             A0(129),A1(129),A2(129),A3(129),
     .             B0(18),B1(18),B2(18),B3(18),
     .             C0(33),C1(33),C2(33),C3(33),
     .             XC(33),XZ(33),XZZ(33),YC(33),YZ(33),YZZ(33),
     .             NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,SCAL,SCALZ,
     .             ALPHA,FMACH,HINGE,OMEGA,PSI,CA,SA,CAC,CAS,
     .             TMACH,TILT,RAD,PI,NIT,PSIS,
     .             AA0,K2,K3,LX,MX,MY,MZ,DX,DXX,WATX,WATY
*COMDECK A
      COMMON/A/    SX(129,33),SZ(129,33),
     .             SXX(129,33),SXZ(129,33),SZZ(129,33)
*COMDECK FLO
      COMMON/FLO/ P1,P2,P3,BETA,FR,IR,JR,KR,GD,IG,JG,KG,NS
```

```
*DECK TFAR1
      PROGRAM TFAR1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     .                 TAPE14,TAPE16,TAPE20)
C     THREE DIMENSIONAL ROTOR BLADE ANALYSIS IN TRANSONIC FLOW
C     USING SHEARED PARABOLIC COORDINATES
C     G IS THE REDUCED VELOCITY POTENTIAL IN THE MOVING FRAME
C     A ROTATIONAL FLOW VERSION OF JAMESON'S FLO17 FOR A ROTOR BLADE
C     PROGRAMMED BY I-CHUNG CHANG, APRIL 1983
*CALL BLANK
*CALL A
*CALL FLO
      DIMENSION    XS(200,33),YS(200,33),
     1             ZS(33),XLE(33),YLE(33),SLOPT(33),TRAIL(33),NP(33),
     2             E1(33),E2(33),E3(33),E4(33),E5(33),
     3             XP(200),YP(200),D1(200),D2(200),D3(200),
     4             X(129),Y(129),SU(129),SV(129),SW(129),SM(129),
     5             CP(129),CHORD(33),SCL(33),SCD(33),SCM(33),
     6             FIT(3),COV0(3),P10(3),P20(3),P30(3),BETA0(3),
     7             FHALF(3),FITMIN(3),KPLOTS(26),TYTLE(10)
      ND         = 200
      NE         = 129
      IREAD      = 5
      IWRIT      = 6
      PI         = 3.14159265358979
      RAD        = 57.2957795130823
      DRAD       = 1./RAD
```

```
 1 WRITE  (IWRIT,600)
   WRITE  (IWRIT,2)
 2 FORMAT(14H0PROGRAM  TFAR1,70X,31H I-CHUNG CHANG,NASA-AMES CENTER/
  1       42H0THREE DIMENSIONAL ROTOR BLADE ANALYSIS IN,
  2       51H TRANSONIC FLOW USING SHEARED PARABOLIC COORDINATES)
   READ(IREAD,530) TYTLE
   WRITE(IWRIT,630) TYTLE
   READ(IREAD,500)
   READ(IREAD,510) FNX,FNY,FNZ
   NX         = FNX
   NY         = FNY
   NZ         = FNZ
   IF (NX.LT.1) GO TO 302
   IPLOT      = -1
   KPLOT      = 1
   READ(IREAD,500)
   NM         = 0
11 NM         = NM  +1
   READ(IREAD,510) FIT(NM),COVO(NM),P10(NM),P20(NM),
  .                P30(NM),BETAO(NM),FHALF(NM)
   IF (FHALF(NM).NE.0..AND.NM.LT.3) GO TO 11
   FHALF(3)   = 0.
   READ(IREAD,500)
   READ(IREAD,510) FSPEED,PSI,ALPHA,TIPWR,RADIUS,AINF
   READ(IREAD,500)
   READ(IREAD,510) CREF,XREF,FBLADE,FCLUST,CD0
   TILT       = ALPHA
   ICLUST     = FCLUST
   ALPHA      = TILT *DRAD
   PSID       = PSI
   PSI        = PSID *DRAD
   DRADIUS    = 1. /RADIUS
   FMACH      = FSPEED/AINF
   TMACH      = TIPWR/AINF
   POMEGA     = TIPWR/(CREF*RADIUS)
   UTREF      = CREF/AINF
   OMEGA      = POMEGA*UTREF
   CA         = FMACH*COS(ALPHA)
   SA         = FMACH*SIN(ALPHA)
   PSIM       = PSI-.5*PI
   CAC        = CA*COS(PSIM)
   CAS        = CA*SIN(PSIM)
   CALL GEOM  (ND,NC,NP,ZS,XS,YS,XLE,YLE,SLOPT,TRAIL,XP,YP,
  1            SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
  2            D1,D2,XTE0,CHORD0,ZTIP,ISYM0,HINGE,FBLADE)
   ISYM       = ISYM0
91 CALL COORD (NX,NY,NZ,XTE0,ZTIP,XMAX,ZMAX,ICLUST,
  1            SY,SCAL,SCALZ,AX,AY,AZ,
  2            A0,A1,A2,A3,B0,B1,B2,B3,C0,C1,C2,C3)
   CALL SINGL (NC,NZ,KTE1,KTE2,CHORD0,
  1            SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
  2            ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
  3            C0,C1,C2,C3,E1,E2,E3,E4,E5,IND)
   CALL SURF  (ND,NC,ZS,SLOPT,TRAIL,XS,YS,NP,
  1            XP,YP,D1,D2,D3,X,Y,IND)
```

```
      IF (IND.EQ.0) GO TO 291
      NM         = 1
      NIT        = 0
      CALL ESTIM
  101 WRITE (IWRIT,600)
      MIT        = FIT(NM)
      KIT        = MIT
      IF (NM.GT.1.AND.FHALF(NM).EQ.0.) KIT = 10
      JIT        = NIT
      KRES       = (MIT  -NIT  -2)/500  +2
      JRES       = 0
      NRES       = 0
      COV        = COVO(NM)
      COU        = 10000000.
      KY         = NY  +1
      K1         = 2
      K2         = NZ
  103 LZ         = NZ/2  +1
      WRITE (IWRIT,104)
  104 FORMAT(49H0INDICATION OF LOCATION OF BLADE AND VORTEX SHEET,
     1       27H IN COORDINATE PLANE Y = 0./
     2       27H0((IV(I,K),K=K1,K2),I=2,NX))
      DO 106 I=2,NX
  106 WRITE (IWRIT,650) (IV(I,K),K=K1,K2)
      WRITE (IWRIT,600)
      WRITE (IWRIT,112)
  112 FORMAT(49H0CHORDWISE CELL DISTRIBUTION IN SQUARE ROOT PLANE,
     1       54H AND MAPPED SURFACE COORDINATES AT CENTER LINE AND TIP/
     2       15H0      A0      ,15H       A1      ,
     3       15H       A2      ,15H       A3      ,
     3       15H   ROOT PROFILE,15H   TIP PROFILE )
      DO 114 I=2,NX
  114 WRITE (IWRIT,610) A0(I),A1(I),A2(I),A3(I),S0(I,KTE1),S0(I,KTE2)
      WRITE (IWRIT,116)
  116 FORMAT(15H0  TE LOCATION ,15H    POWER LAW  )
      WRITE (IWRIT,610) XMAX,AX
      WRITE (IWRIT,600)
      WRITE (IWRIT,118)
  118 FORMAT(46H0NORMAL CELL DISTRIBUTION IN SQUARE ROOT PLANE/
     1       15H0      B0      ,15H       B1      ,
     2       15H       B2      ,15H       B3      )
      DO 120 J=2,KY
  120 WRITE (IWRIT,610) B0(J),B1(J),B2(J),B3(J)
      WRITE (IWRIT,122)
  122 FORMAT(15H0  SCALE FACTOR,15H    POWER LAW  )
      WRITE (IWRIT,610) SY,AY
      WRITE (IWRIT,600)
      WRITE (IWRIT,124)
  124 FORMAT(27H0SPANWISE CELL DISTRIBUTION/
     1       15H0      C0      ,15H       C1      ,15H       C2      ,
     2       15H       C3      )
      DO 126 K=K1,K2
  126 WRITE (IWRIT,610) C0(K),C1(K),C2(K),C3(K)
      WRITE (IWRIT,128)
  128 FORMAT(15H0  TIP LOCATION,15H    POWER LAW  )
```

```
      WRITE (IWRIT,610) ZMAX,AZ
      WRITE (IWRIT,600)
      WRITE (IWRIT,125)
  125 FORMAT(14H0SINGULAR LINE/
     1       15H0      X SING    ,15H       Y SING     ,
     2       15H         XZ     ,15H          YZ     ,15H         XZZ     ,
     3       15H        YZZ      )
      DO 127 K=K1,K2
  127 WRITE (IWRIT,610) XC(K),YC(K),XZ(K),YZ(K),XZZ(K),YZZ(K)
      WRITE (IWRIT,600)
      WRITE (IWRIT,132)
  132 FORMAT(35H0ITERATIVE SOLUTION --- STEADY MODE)
      WRITE (IWRIT,134)
  134 FORMAT(15H0          NX     ,15H          NY     ,15H          NZ     )
      WRITE (IWRIT,640) NX,NY,NZ
      CALL SECOND(T)
      WRITE (IWRIT,700) T
      WRITE (IWRIT,136)
  136 FORMAT(15H0   TMACH NO    ,15H    FMACH NO    ,15H  TITL ANG      ,
     1       15H AZIMUTHAL ANG )
      WRITE (IWRIT,610) TMACH,FMACH,TILT,PSID
      WRITE (IWRIT,138)
  138 FORMAT(10H0ITERATION,15H    CORRECTION ,4H   I ,4H   J ,4H   K ,
     1                     15H      RESIDUAL ,4H   I ,4H   J ,4H   K ,
     2       10H CIRCULATN,10H REL FCT 1,10H REL FCT 2,10H REL FCT 3,
     3       10H    BETA  ,10H SONIC PTS)
  141 NIT        = NIT  +1
      JIT        = JIT  +1
      P1         = P10(NM)
      P2         = P20(NM)
      P3         = P30(NM)
      BETA       = BETA0(NM)
      CALL RELAX
      WRITE (IWRIT,660) NIT,GD,IG,JG,KG,FR,IR,JR,KR,E0(LZ),
     1                  P1,P2,P3,BETA,NS
      IF (NIT.LT.MIT.AND.GD.GT.COV.AND.GD.LT.10.) GO TO 141
      IF(FHALF(NM).EQ.0.) GO TO 176
      NM         = NM   +1
      NX         = NX   +NX
      NY         = NY   +NY
      NZ         = NZ   +NZ
      CALL COORD (NX,NY,NZ,XTE0,ZTIP,XMAX,ZMAX,ICLUST,
     1           SY,SCAL,SCALZ,AX,AY,AZ,
     2           A0,A1,A2,A3,B0,B1,B2,B3,C0,C1,C2,C3)
      CALL SINGL (NC,NZ,KTE1,KTE2,CHORD0,
     1           SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
     2           ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
     3           C0,C1,C2,C3,E1,E2,E3,E4,E5,IND)
      CALL SURF  (ND,NC,ZS,SLOPT,TRAIL,XS,YS,NP,
     1           XP,YP,D1,D2,D3,X,Y,IND)
      CALL REFIN
      NIT        = 0
      GO TO 101
  176 LX         = NX/2   +1
      K          = 2
```

```
      WRITE(IWRIT,600)
      WRITE(IWRIT,184) PSID
  184 FORMAT(1H0,*AZIMUTHAL ANGLE = *,F15.5)
  171 K           = K   +1
      IF (K.EQ.MZ) GO TO 191
      IF (K.LT.KTE1.OR.K.GT.KTE2) GO TO 171
      I1          = ITE1(K)
      I2          = ITE2(K)
      ZSEC        = CO(K) + HINGE
      VROT        = OMEGA*ZSEC
      VTAN        = VROT + FMACH*SIN(PSI)*COS(ALPHA)
      SMACH(K)    = VTAN
      CALL VELO(K,SU,SV,SW,SM,CP,X,Y)
  175 CHORD(K)    = X(I1)   -X(LX)
      CALL FORCF(I1,I2,X,Y,CP,TILT,CHORD(K),XC(K),SCL(K),SCD(K),SCM(K))
      CALL PSURE(IPLOT,K,X,Y,CP,I1,I2,SCL(K),SCD(K),SCM(K))
      WRITE (IWRIT,600)
      WRITE (IWRIT,182)
  182 FORMAT(24H0SECTION CHARACTERISTICS/
     1       15H0  SPAN STATION,15H        CL     ,15H        CD     ,
     2       15H         CM     )
      ZPHYS= CO(K) + HINGE
  185 WRITE (IWRIT,610)ZPHYS,SCL(K),SCD(K),SCM(K)
      IF (KPLOT.GE.0) CALL CPLOT (I1,I2,SMACH(K),X,Y,SU,SV,SW,SM,CP)
      GO TO 171
  191 CALL TOTFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,CO,XC,
     1            CL,CD1,CMP,CMR,CMY)
      CD          = CD0   +CD1
      VLD1        = 0.
      IF (ABS(CD1).GT.1.E-6) VLD1 = CL/CD1
      VLD         = 0.
      IF (ABS(CD).GT.1.E-6) VLD = CL/CD
      WRITE (IWRIT,600)
      WRITE (IWRIT,192)
  192 FORMAT(22H0BLADE CHARACTERISTICS/
     1       15H0        CL     ,15H     CD FORM   ,15H   CD FRICTION ,
     2       15H         CD     ,15H     L/D FORM  ,15H        L/D     )
      WRITE (IWRIT,610) CL,CD1,CD0,CD,VLD1,VLD
      WRITE (IWRIT,196)
  196 FORMAT(15H0     CM PITCH  ,15H     CM ROLL   ,15H      CM YAW    )
      WRITE (IWRIT,610) CMP,CMR,CMY
  210 CALL THREED(IPLOT,SU,SV,SW,SM,CP,X,Y,TYTLE,CHORD,
     .            CL,CD,CHORD0,SCL,SCD,SCM)
      CALL ROTORB(IPLOT,SU,SV,SW,SM,CP,X,Y)
      GO TO 301
  291 WRITE (IWRIT,600)
      WRITE (IWRIT,292)
  292 FORMAT(24H0BAD DATA,SPLINE FAILURE)
  301 IF(IPLOT.EQ.0) CALL DONEPL
  302 STOP
  500 FORMAT(1X)
  510 FORMAT(8F10.6)
  511 FORMAT(26I3)
  530 FORMAT(10A8)
  600 FORMAT(1H1)
```

```
  610 FORMAT(F12.4,7F15.4)
  620 FORMAT(8E15.5)
  630 FORMAT(1H0,10A8)
  640 FORMAT(I8,7I15)
  650 FORMAT(1X,32I4)
  660 FORMAT(I10,E15.5,3I4,E15.5,3I4,5F10.5,I10)
  661 FORMAT(I10,E15.5,3I4,E15.5,3I4,I10)
  670 FORMAT(2E15.4,2F15.4)
  700 FORMAT(15H0COMPUTING TIME,F10.3,10H   SECONDS)
  900 FORMAT(1X,F12.8,5X,F12.8,5X,F12.8,5X,F12.8,5X,F12.8,5X,
     .       F12.8,5X,F12.8)
      END
```

```
*DECK GEOM
      SUBROUTINE GEOM  (ND,NC,NP,ZS,XS,YS,XLE,YLE,SLOPT,TRAIL,XP,YP,
     1                  SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
     2                  D1,D2,XTE0,CHORD0,ZTIP,ISYM0,HINGE,FBLADE)
C     GEOMETRIC DEFINITION OF ROTOR BLADE
      DIMENSION   XS(ND,1),YS(ND,1),ZS(1),XLE(1),YLE(1),D1(1),D2(1),
     1            SLOPT(1),TRAIL(1),XP(1),YP(1),NP(1)
      IREAD     = 5
      IWRIT     = 6
      RAD       = 57.2957795130823
      READ(IREAD,500)
      READ(IREAD,510) FNC,SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED
      IF (FNC.LT.3.) RETURN
      NC        = FNC
      WRITE (IWRIT,2)
    2 FORMAT(15H0    SWEEP(1)  ,15H     SWEEP(2)  ,15H   FINAL SWEEP ,
     1       15H     DIHED(1)  ,15H     DIHED(2)  ,15H   FINAL DIHED )
      WRITE (IWRIT,610) SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED
      SWEEP1    = SWEEP1/RAD
      SWEEP2    = SWEEP2/RAD
      SWEEP     = SWEEP/RAD
      DIHED1    = DIHED1/RAD
      DIHED2    = DIHED2/RAD
      DIHED     = DIHED/RAD
      ISYM0     = 1
      XTE0      = 0.
      CHORD0    = 0.
      K         = 1
   11 READ(IREAD,500)
      READ(IREAD,510) ZS(K),XL,YL,CHORD,THICK,TWIST,FSEC
      AL        = TWIST
      ALPHA     = AL/RAD
      IF (FSEC.EQ.0.) GO TO 31
      READ  (IREAD,500)
      READ  (IREAD,510) YSYM,FNU,FNL
      NU        = FNU
      NL        = FNL
```

```
      N          = NU   +NL   -1
      READ   (IREAD,500)
      DO 12 I=NL,N
   12 READ   (IREAD,510) XP(I),YP(I)
      L          = NL   +1
      IF (YSYM.GT.0.) GO TO 15
      READ   (IREAD,500)
      DO 8 I=1,NL
      READ   (IREAD,510) VAL,DUM
      J          = L   -I
      XP(J)      = VAL
    8 YP(J)      = DUM
      GO TO 21
   15 J          = L
      DO 16 I=NL,N
      J          = J   -1
      XP(J)      = XP(I)
   16 YP(J)      = -YP(I)
   21 WRITE (IWRIT,600)
      WRITE (IWRIT,22) ZS(K)
   22 FORMAT(16H0PROFILE AT Z = ,F10.5/
     1       15H0    TE ANGLE   ,15H     TE SLOPE   ,15H        X SING   ,
     2       15H     Y SING    )
      CALL SINGPT(XP,YP,NL,N,XSING,YSING,TRL,SLT)
      WRITE (IWRIT,610) TRL,SLT,XSING,YSING
      WRITE (IWRIT,24)
   24 FORMAT(15H0        X      ,15H         Y      )
      DO 26 I=1,N
   26 WRITE (IWRIT,610) XP(I),YP(I)
   31 SCALE      = CHORD/(XP(1)   -XP(NL))
      DO 33 I=1,N
      D1(I)      = XL   + SCALE*(XP(I)-XP(NL))
   33 D2(I)      = YL   + SCALE*(YP(I)-YP(NL))*THICK
      CALL SINGPT(D1,D2,NL,N,XSING,YSING,TRL,SLT)
      XLE(K)     = XSING
      YLE(K)     = YSING
      CA         = COS(ALPHA)
      SA         = SIN(ALPHA)
      DO 32 I=1,N
      XS(I,K)    = (D1(I)   -XSING)*CA   +(D2(I)   -YSING)*SA
   32 YS(I,K)    = (D2(I)   -YSING)*CA   -(D1(I)   -XSING)*SA
      SLOPT(K)   = SLT -TAN(ALPHA)
      TRAIL(K)   = TRL/RAD
      NP(K)      = N
      XTEO       = AMAX1(XTEO,XS(1,K))
      CHORDO     = AMAX1(CHORDO,CHORD)
      WRITE (IWRIT,52) ZS(K)
   52 FORMAT(27H0SECTION DEFINITION AT Z = ,F10.5/
     1       15H0      XLE      ,15H       YLE      ,15H      CHORD     ,
     2       15HTHICKNESS RATIO,15H  TWIST ANGLE  )
      WRITE (IWRIT,610) XL,YL,CHORD,THICK,AL
      K          = K   +1
      IF (K.LE.NC) GO TO 11
   65 ZO         = (1.-.5*FBLADE)*(ZS(NC)-ZS(1)) +ZS(1)
      KK         = 0
```

```
      ZTIP         = ZS(NC) -Z0
      DO 63 K=1,NC
      ZS(K)        = ZS(K) -Z0
      IF(ABS(ZS(K)).GT.ZTIP) GO TO 63
      KK           = KK +1
      N            = NP(K)
      DO 64 I=1,N
      XS(I,KK)     = XS(I,K)
      YS(I,KK)     = YS(I,K)
   64 CONTINUE
      ZS(KK)       = ZS(K)
      XLE(KK)      = XLE(K)
      YLE(KK)      = YLE(K)
      SLOPT(KK)    = SLOPT(K)
      TRAIL(KK)    = TRAIL(K)
      NP(KK)       = NP(K)
   63 CONTINUE
      NC           = KK
      HINGE        = Z0
      RETURN
      RETURN
  500 FORMAT(1X)
  510 FORMAT(8F10.6)
  600 FORMAT(1H1)
  610 FORMAT(F12.4,7F15.4)
      END
```

```
*DECK COORD
      SUBROUTINE COORD (NX,NY,NZ,XTEO,ZTIP,XMAX,ZMAX,ICLUST,
     1                  SY,SCAL,SCALZ,AX,AY,AZ,
     2                  A0,A1,A2,A3,B0,B1,B2,B3,C0,C1,C2,C3)
C     SETS UP STRETCHED PARABOLIC AND SPANWISE COORDINATES
      DIMENSION    A0(1),A1(1),A2(1),A3(1),B0(1),B1(1),B2(1),B3(1),
     1             C0(1),C1(1),C2(1),C3(1)
      PI           = 3.14159265358979
      DX           = 2./NX
      DY           = 1./NY
      DZ           = 2./NZ
      DDX          = 1./DX
      DDXX         = DDX*DDX
      DDY          = 1./DY
      DDZ          = 1./DZ
      KY           = NY +1
      AX           = .5
      AY           = .5
      AZ           = .5
      XMAX         = .625
      ZMAX         = .625
      SY           = .5
      SCAL         = XTEO/(.50001*XMAX*XMAX)
```

```
      SCALZ       = ZTIP/(1.000001*ZMAX)
      W1          = SCAL/SCALZ
      U2          = 1
      V2          = (DX*DDY)**2
      W2          = (DX*W1*DDZ)**2
      DO 12 I=2,NX
      DD          = (I   -1)*DX   -1.
      B           = 1.
      IF (ABS(DD).GT.XMAX) GO TO 13
      D0          = DD
      D1          = 1.
      D2          = 0.
      GO TO 8
   13 IF (DD.LT.0.) B = -1.
      A           = 1.-(DD-B*XMAX)**2
      C           = A**AX
      D           = (AX   +AX   -1.)*(1.   -A)
      D0          = B*XMAX+(DD-B*XMAX)/C
      D1          = A*C/(1.+D)
      D2          = -2.*AX*(DD-B*XMAX)*(3.+D)/((1.+D)*A)
    8 A0(I)       = D0
      A1(I)       = .5*D1*DDX
      A2(I)       = D1*D1*U2
   12 A3(I)       = .5*DX*D2
      DO 22 J=2,KY
      DD          = (J-2) *DY
      A           = 1.   -DD*DD
      C           = A**AY
      D           = (AY   +AY   -1.)*(1.   -A)
      D1          = A*C/((1.   +D)*SY)
      B0(J)       = SY*DD/C
      B1(J)       = .5*D1*DDY
      B2(J)       = D1*D1*V2
   22 B3(J)       = -AY*DD*DY*(3.   +D)/((1.   +D)*A)
      IF(ICLUST.EQ.0) GO TO 30
      AH          = .049
      BH          = AH/7.
      CH          = 8.*PI
      DH          = PI/7.
      EH          = 8.*DH
   30 DO 32 K=2,NZ
      DD          = (K   -1)*DZ   -1.
      B           = 1.
      IF (ABS(DD).GT.ZMAX) GO TO 33
      IF(ICLUST.NE.0) GO TO 40
      D0          = DD
      D1          = 1.
      D2          = 0.
      GO TO 34
   40 DD          = .8*(DD + ZMAX)
      IF(DD.GT..125) GO TO 45
      A           = CH*DD
      B           = BH*SIN(A)
      D0          = DD -B
      D1          = 1./(1.- CH*BH*COS(A))
```

```
      D2            = -D1*CH*CH*B
      GO TO 46
   45 A             = (8.*DD - 1.)*DH
      B             = AH*SIN(A)
      D0            = DD + B
      D1            = 1./(1.+ AH*EH*COS(A))
      D2            = D1*EH*EH*B
   46 D0            = 1.25*D0-ZMAX
      GO TO 34
   33 IF (DD.LT.0.) B = -1.
      A             = 1.-(DD-B*ZMAX)**2
      C             = A**AZ
      D             = (AZ  +AZ  -1.)*(1.  -A)
      D0            = B*ZMAX+(DD-B*ZMAX)/C
      D1            = A*C/(1.+D)
      D2            = -2.*AZ*(DD-B*ZMAX)*(3.+D)/((1.+D)*A)
   34 C0(K)         = SCALZ*D0
      C1(K)         = .5*D1*W1*DDZ
      C2(K)         = D1*D1*W2
   32 C3(K)         = .5*DZ*D2
      RETURN
      END
```

```
*DECK SINGL
      SUBROUTINE SINGL (NC,NZ,KTE1,KTE2,CHORD0,
     1                  SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
     2                  ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
     3                  C0,C1,C2,C3,E1,E2,E3,E4,E5,IND)
C     GENERATES SINGULAR LINE FOR SQUARE ROOT TRANSFORMATION
      DIMENSION ZS(1),XLE(1),YLE(1),XC(1),XZ(1),XZZ(1),
     1          YC(1),YZ(1),YZZ(1),C0(1),C1(1),C2(1),C3(1),
     2          E1(1),E2(1),E3(1),E4(1),E5(1)
      DO 2 K=1,NC
      E4(K)         = 0.
    2 E5(K)         = 0.
      K2            = NZ
   11 DO 12 K=2,K2
      IF (C0(K).LT.ZS(1)) KTE1 = K +1
      IF (C0(K).LE.ZS(NC)) KTE2 = K
   12 CONTINUE
      B             = CHORD0
      S1            = TAN(SWEEP1)
      S2            = TAN(SWEEP2)
      T1            = TAN(DIHED1)
      T2            = TAN(DIHED2)
      CALL SPLIF (1,NC,ZS,XLE,E1,E2,E3,1,S1,1,S2,0,0.,IND)
      CALL INTPL (KTE1,KTE2,C0,XC,1,NC,ZS,XLE,E1,E2,E3,0)
      CALL INTPL (KTE1,KTE2,C0,XZ,1,NC,ZS,E1,E2,E3,E4,0)
      CALL INTPL (KTE1,KTE2,C0,XZZ,1,NC,ZS,E2,E3,E4,E5,0)
```

```
      CALL SPLIF (1,NC,ZS,YLE,E1,E2,E3,1,T1,1,T2,0,0.,IND)
      CALL INTPL (KTE1,KTE2,C0,YC,1,NC,ZS,YLE,E1,E2,E3,0)
      CALL INTPL (KTE1,KTE2,C0,YZ,1,NC,ZS,E1,E2,E3,E4,0)
      CALL INTPL (KTE1,KTE2,C0,YZZ,1,NC,ZS,E2,E3,E4,E5,0)
      S           = B*TAN(SWEEP)
      S1          = B*S1
      S2          = B*S2
      T           = B*TAN(DIHED)
      T1          = B*T1
      T2          = B*T2
      N           = KTE1   -1
      DO 22 K=2,N
      ZZ          = (C0(K)   -C0(KTE1))/B
      A           = EXP(ZZ)
      XC(K)       = XC(KTE1)   +S*ZZ   -(S1   -S)*(1.   -A)
      YC(K)       = YC(KTE1)   +T*ZZ   -(T1   -T)*(1.   -A)
      XZ(K)       = (S   +(S1   -S)*A)/B
      YZ(K)       = (T   +(T1   -T)*A)/B
      XZZ(K)      = (S1   -S)*A/(B*B)
   22 YZZ(K)      = (T1   -T)*A/(B*B)
   31 N           = KTE2   +1
      DO 32 K=N,K2
      ZZ          = (C0(K)   -C0(KTE2))/B
      A           = EXP(-ZZ)
      XC(K)       = XC(KTE2)   +S*ZZ   +(S2   -S)*(1.   -A)
      YC(K)       = YC(KTE2)   +T*ZZ   +(T2   -T)*(1.   -A)
      XZ(K)       = (S   +(S2   -S)*A)/B
      YZ(K)       = (T   +(T2   -T)*A)/B
      XZZ(K)      = -(S2   -S)*A/(B*B)
   32 YZZ(K)      = -(T2   -T)*A/(B*B)
      RETURN
      END
```

```
*DECK SURF
      SUBROUTINE SURF(ND,NC,ZS,SLOPT,TRAIL,XS,YS,NP,
     1                XP,YP,D1,D2,D3,X,Y,IND)
C     INTERPOLATES MAPPED WING SURFACE AT MESH POINTS
C     INTERPOLATION IS LINEAR IN PHYSICAL PLANE
*CALL BLANK
*CALL A
      DIMENSION   XS(ND,1),YS(ND,1),ZS(1),SLOPT(1),TRAIL(1),X(1),Y(1),
     1            XP(1),YP(1),D1(1),D2(1),D3(1),NP(1)
      PI          = 3.14159265358979
      DX          = 2./NX
      LX          = NX/2   +1
      MX          = NX   +1
      MZ          = NZ   +1
      IV0         = 1
      IV1         = -1
      DO 2 K=1,MZ
```

```
      ITE1(K)    = MX
      ITE2(K)    = MX
      DO 2 I=1,MX
      IV(I,K)    = -2
    2 SO(I,K)    = 0.
      K          = KTE1
      K2         = 1
   21 K2         = K2   +1
      K1         = K2   -1
      R2         = 1.
      IF (ZS(K2)   -CO(K)) 21,25,23
   23 R2         = (CO(K)   -ZS(K1))/(ZS(K2)   -ZS(K1))
   25 R1         = 1.   -R2
      C          = R1*XS(1,K1)   +R2*XS(1,K2)
      CC         = SQRT((C   +C)/SCAL)
      DO 32 I=2,NX
      IF (AO(I)   .LT.-CC) I1 = I   +1
      IF (AO(I)   .LE.CC) I2 = I
   32 CONTINUE
      ITE1(K)    = I1
      ITE2(K)    = I2
      CC         = AO(I2)/CC
      KK         = K1
      P          = R1
   41 N          = NP(KK)
      Q          = SQRT(XS(1,KK)/C)/CC
      DO 42 I=2,NX
   42 X(I)       = Q*AO(I)
      ANGL       = PI   +PI
      U          = 1.
      V          = 0.
      DO 44 I=1,N
      R          = SQRT(XS(I,KK)**2   +YS(I,KK)**2)
      IF (R.EQ.0.) GO TO 45
      ANGL       = ANGL   +ATAN2((U*YS(I,KK)   -V*XS(I,KK)),
     1                          (U*XS(I,KK)   +V*YS(I,KK)))
      U          = XS(I,KK)
      V          = YS(I,KK)
      R          = SQRT((R   +R)/SCAL)
      XP(I)      = R*COS(.5*ANGL)
      YP(I)      = R*SIN(.5*ANGL)
      GO TO 44
   45 ANGL       = PI
      U          = -1.
      V          = 0.
      XP(I)      = 0.
      YP(I)      = 0.
   44 CONTINUE
      ANGL       = ATAN(SLOPT(KK))
      ANGL1      = ATAN(YS(1,KK)/XS(1,KK))
      ANGL2      = ATAN(YS(N,KK)/XS(N,KK))
      ANGL1      = ANGL   -.5*(ANGL1   -TRAIL(KK))
      ANGL2      = ANGL   -.5*(ANGL2   +TRAIL(KK))
      T1         = TAN(ANGL1)
      T2         = TAN(ANGL2)
```

```
      CALL SPLIF (1,N,XP,YP,D1,D2,D3,1,T1,1,T2,0,0.,IND)
      CALL INTPL (I1,I2,X,Y,1,N,XP,YP,D1,D2,D3,0)
      X1          = .25*XS(1,KK)
      A           = SLOPT(KK)*(XS(1,KK)   -X1)
      B           = 1./(XS(1,KK)   -X1)
      ANGL        = PI   +PI
      U           = 1.
      V           = 0.
      M           = I1   -1
      DO 52 I=2,M
      XX          = .5*SCAL*X(I)**2
      D           = B*(XX   -X1)
      YY          = YS(1,KK)   +A*ALOG(D)/D
      R           = SQRT(XX**2   +YY**2)
      ANGL        = ANGL   +ATAN2((U*YY   -V*XX),(U*XX   +V*YY))
      U           = XX
      V           = YY
      R           = SQRT((R   +R)/SCAL)
   52 Y(I)        = R*SIN(.5*ANGL)
      A           = SLOPT(KK)*(XS(N,KK)   -X1)
      B           = 1./(XS(N,KK)   -X1)
      ANGL        = 0.
      U           = 1.
      V           = 0.
      M           = I2   +1
      DO 54 I=M,NX
      XX          = .5*SCAL*X(I)**2
      D           = B*(XX   -X1)
      YY          = YS(N,KK)   +A*ALOG(D)/D
      R           = SQRT(XX**2   +YY**2)
      ANGL        = ANGL   +ATAN2((U*YY   -V*XX),(U*XX   +V*YY))
      U           = XX
      V           = YY
      R           = SQRT((R   +R)/SCAL)
   54 Y(I)        = R*SIN(.5*ANGL)
      Q           = P/(Q*CC)
      DO 62 I=2,NX
   62 SO(I,K)     = SO(I,K)   +Q*Y(I)
      IF (KK.EQ.K2) GO TO 71
      KK          = K2
      P           = R2
      GO TO 41
   71 DO 72 I=I1,I2
   72 IV(I,K)     = 2
      M           = I1   -1
      DO 74 I=2,M
      ZZ          = CO(K)
      IF (ZZ.GE.CO(KTE1)) IV(I,K) = IVO
   74 CONTINUE
      M           = I2   +1
      DO 76 I=M,NX
      ZZ          = CO(K)
      IF (ZZ.GE.CO(KTE1)) IV(I,K) = IVO
   76 CONTINUE
      K2          = K2   -1
```

```
      K             = K   +1
      IF (K.LE.KTE2) GO TO 21
      K1            = 2
      K2            = NZ
   81 DO 82 I=2,NX
      ZZ            = CO(K)
      IF (ZZ.LE.ZS(NC).AND.ZZ.GE.CO(KTE1)) IV(I,K) = IV0
   82 CONTINUE
      K             = K   +1
      IF (K.LE.K2) GO TO 81
      DO 102 K=K1,K2
      DO 104 I=2,NX
      IF (IV(I,K).GT.0) GO TO 104
      IF (IV(I+1,K+1).GT.0.OR.IV(I-1,K+1).GT.0) IV(I,K) = IV1
      IF (IV(I+1,K-1).GT.0.OR.IV(I-1,K-1).GT.0) IV(I,K) = IV1
  104 CONTINUE
  102 IF (SO(LX,K).LT.1.E-05) IV(LX,K) = 0
      DO 13 K=2,NZ
      DO 13 I=2,NX
      SI            = SO(I+1,K)   -SO(I-1,K)
      SK            = SO(I,K+1)   -SO(I,K-1)
      SX(I,K)       = A1(I)* SI
      SZ(I,K)       = C1(K)* SK
      SXX(I,K)      = (SO(I+1,K)-2.*SO(I,K)+SO(I-1,K)+A3(I)*SI)*A2(I)
      SZZ(I,K)      = (SO(I,K+1)-2.*SO(I,K)+SO(I,K-1)+C3(K)*SK)*C2(K)
   13 SXZ(I,K)      = (SO(I+1,K+1)-SO(I-1,K+1)-SO(I+1,K-1)+SO(I-1,K-1))
     .                *A1(I)*C1(K)*DX*DX
      RETURN
      END



*DECK SINGPT
      SUBROUTINE SINGPT(X,Y,NL,N,XSING,YSING,TRL,SLT)
      DIMENSION X(1),Y(1)
      RAD           = 57.29577951308232
      NP            = NL+1
      NM            = NL-1
      CALL XYSING(X(NL),Y(NL),X(NP),Y(NP),X(NM),Y(NM),XSING,YSING)
      SLOPU       = (Y(N)-Y(N-1))/(X(N)-X(N-1))
      SLOPL       = (Y(1)-Y(2))/(X(1)-X(2))
      SLT         = .5*(SLOPU +SLOPL)
      THETAU      = ATAN2(SLOPU,1.)*RAD
      THETAL      = ATAN2(SLOPL,1.)*RAD
      TRL         = THETAL -THETAU
      RETURN
      END
```

```
*DECK XYSING
      SUBROUTINE XYSING (X1,Y1,X2,Y2,X3,Y3,XSING,YSING)
C     FITS CIRCLE TO 3 POINTS NEAR LEADING EDGE AND FIND THE CENTER
      YA   = (Y2 + Y1)*.5E0
      XA   = (X2 + X1)*.5E0
      YB   = (Y3 + Y1)*.5E0
      XB   = (X3 + X1)*.5E0
      SL1 = -(X2 - X1) / (Y2 - Y1)
      SL2 = -(X3 - X1) / (Y3 - Y1)
      XSING2 = (SL1 * XA - SL2 * XB + YB - YA) / (SL1 - SL2)
      XSING  = (XSING2 + X1)*.5E0
      YSING2 = SL1 * (XSING2 -XA) + YA
      YSING  = (YSING2 + Y1)*.5E0
      RETURN
      END
```

```
*DECK SPLIF
      SUBROUTINE SPLIF(M,N,S,F,FP,FPP,FPPP,KM,VM,KN,VN,MODE,FQM,IND)
C     SPLINE FIT - JAMESON
C     INTEGRAL PLACED IN FPPP IF MODE GREATER THAN 0
C     IND SET TO ZERO IF DATA ILLEGAL
      DIMENSION   S(1),F(1),FP(1),FPP(1),FPPP(1)
      IND         = 0
      K           = IABS(N  -M)
      IF (K  -1) 81,81,1
    1 K           = (N  -M)/K
      I           = M
      J           = M  +K
      DS          = S(J)  -S(I)
      D           = DS
      IF (DS) 11,81,11
   11 DF          = (F(J)  -F(I))/DS
      IF (KM  -2) 12,13,14
   12 U           = .5
      V           = 3.*(DF  -VM)/DS
      GO TO 25
   13 U           = 0.
      V           = VM
      GO TO 25
   14 U           = -1.
      V           = -DS*VM
      GO TO 25
   21 I           = J
      J           = J  +K
      DS          = S(J)  -S(I)
      IF (D*DS) 81,81,23
   23 DF          = (F(J)  -F(I))/DS
      B           = 1./(DS  +DS  +U)
      U           = B*DS
      V           = B*(6.*DF  -V)
```

```
   25 FP(I)       = U
      FPP(I)      = V
      U           = (2.   -U)*DS
      V           = 6.*DF   +DS*V
      IF (J   -N) 21,31,21
   31 IF (KN   -2) 32,33,34
   32 V           = (6.*VN   -V)/U
      GO TO 35
   33 V           = VN
      GO TO 35
   34 V           = (DS*VN   +FPP(I))/(1.   +FP(I))
   35 B           = V
      D           = DS
   41 DS          = S(J)   -S(I)
      U           = FPP(I)   -FP(I)*V
      FPPP(I)     = (V   -U)/DS
      FPP(I)      = U
      FP(I)       = (F(J)   -F(I))/DS   -DS*(V   +U   +U)/6.
      V           = U
      J           = I
      I           = I   -K
      IF (J   -M) 41,51,41
   51 I           = N   -K
      FPPP(N)     = FPPP(I)
      FPP(N)      = B
      FP(N)       = DF   +D*(FPP(I)   +B   +B)/6.
      IND         = 1
      IF (MODE) 81,81,61
   61 FPPP(J)     = FQM
      V           = FPP(J)
   71 I           = J
      J           = J   +K
      DS          = S(J)   -S(I)
      U           = FPP(J)
      FPPP(J)     = FPPP(I)   +.5*DS*(F(I)   +F(J)   -DS*DS*(U   +V)/12.)
      V           = U
      IF (J   -N)   71,81,71
   81 RETURN
      END
```

```
*DECK INTPL
      SUBROUTINE INTPL(MI,NI,SI,FI,M,N,S,F,FP,FPP,FPPP,MODE)
C     INTERPOLATION USING TAYLOR SERIES - JAMESON
C     ADDS CORRECTION FOR PIECEWISE CONSTANT FOURTH DERIVIATIVE
C     IF MODE GREATER THAN 0
      DIMENSION   SI(1),FI(1),S(1),F(1),FP(1),FPP(1),FPPP(1)
      K           = IABS(N   -M)
      K           = (N   -M)/K
      I           = M
      MIN         = MI
```

```
      NIN          = NI
      D            = S(N)   -S(M)
      IF (D*(SI(NI)   -SI(MI))) 11,13,13
   11 MIN          = NI
      NIN          = MI
   13 KI           = IABS(NIN   -MIN)
      IF (KI) 21,21,15
   15 KI           = (NIN   -MIN)/KI
   21 II           = MIN   -KI
      C            = 0.
      IF (MODE) 31,31,23
   23 C            = 1.
   31 II           = II   +KI
      SS           = SI(II)
   33 I            = I   +K
      IF (I   -N) 35,37,35
   35 IF (D*(S(I)   -SS)) 33,33,37
   37 J            = I
      I            = I   -K
      SS           = SS   -S(I)
      FPPPP        = C*(FPPP(J)   -FPPP(I))/(S(J)   -S(I))
      FF           = FPPP(I)   +.25*SS*FPPPP
      FF           = FPP(I)   +SS*FF/3.
      FF           = FP(I)   +.5*SS*FF
      FI(II)       = F(I)   +SS*FF
      IF (II   -NIN) 31,41,31
   41 RETURN
      END
```

```
*DECK CPLOT
      SUBROUTINE CPLOT (I1,I2,FMACH,X,Y,SU,SV,SW,SM,CP)
C     PLOTS CP AT EQUAL INTERVALS IN THE MAPPED PLANE
      DIMENSION    KODE(3),LINE(75),X(1),Y(1),SU(1),SV(1),SW(1),
     .             SM(1),CP(1)
      DATA         KODE/1H ,1H+,1H*/
      IWRIT        = 6
      WRITE (IWRIT,2)
    2 FORMAT(50H0PLOT OF CP AT EQUAL INTERVALS IN THE MAPPED PLANE/
     1       8H0    X    ,8H     Y    ,8H     SU    ,
     2       8H     SV   ,8H     SW   ,8H     SM    ,8H    CP     )
      FMACH2       = FMACH*FMACH
      AA0          = (1.+.2*FMACH2)
      CP0          = (AA0**3.5 -1.)/(.7*FMACH2)
      AAC          = (1.+.2*FMACH2)/1.2
      CPC          = (AAC**3.5 -1.)/(.7*FMACH2)
      DO 12 I=1,75
   12 LINE(I)      = KODE(1)
      DO 22 I=I1,I2
      KC           = 20.*(CP0   -CPC)   +20.
      KC           = MAX0(1,KC)
```

```
      KC          = MINO(75,KC)
      KK          = 20.*(CP0   -CP(I)) +20.
      KK          = MAXO(1,KK)
      KK          = MINO(75,KK)
      LINE(KC)    = KODE(3)
      LINE(KK)    = KODE(2)
      WRITE(IWRIT,610)X(I),Y(I),SU(I),SV(I),SW(I),
     .                    SM(I),CP(I),LINE
      LINE(KC)    = KODE(1)
   22 LINE(KK)    = KODE(1)
      RETURN
  610 FORMAT(1H ,7F8.3,75A1)
      END
```

```
*DECK FORCF
      SUBROUTINE FORCF (I1,I2,X,Y,CP,AL,CHORD,XM,CL,CD,CM)
C     CALCULATES SECTION FORCE COEFFICIENTS
      DIMENSION    X(1),Y(1),CP(1)
      RAD         = 57.2957795130823
      ALPHA       = AL/RAD
      CL          = 0.
      CD          = 0.
      CM          = 0.
      N           = I2   -1
      DO 12 I=I1,N
      DX          = (X(I+1)   -X(I))/CHORD
      DY          = (Y(I+1)   -Y(I))/CHORD
      XA          = (.5*(X(I+1)   +X(I))   -XM)/CHORD
      YA          = .5*(Y(I+1)   +Y(I))/CHORD
      CPA         = .5*(CP(I+1)   +CP(I))
      DCL         = -CPA*DX
      DCD         = CPA*DY
      CL          = CL   +DCL
      CD          = CD   +DCD
   12 CM          = CM   +DCD*YA   -DCL*XA
      DCL         = CL*COS(ALPHA)   -CD*SIN(ALPHA)
      CD          = CL*SIN(ALPHA)   +CD*COS(ALPHA)
      CL          = DCL
      RETURN
      END
```

```
*DECK TOTFOR
      SUBROUTINE TOTFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,C0,XC,
     1                      CL,CD,CMP,CMR,CMY)
C     CALCULATES TOTAL FORCE COEFFICIENTS
```

```
      DIMENSION   CHORD(1),SCL(1),SCD(1),SCM(1),CO(1),XC(1)
      SPAN        = CO(KTE2)  -CO(KTE1)
      CL          = 0.
      CD          = 0.
      CMP         = 0.
      CMR         = 0.
      CMY         = 0.
      S           = 0.
      N           = KTE2  -1
      DO 12 K=KTE1,N
      DZ          = .5*(CO(K+1)  -CO(K))
      AZ          = .5*(CO(K+1)  +CO(K))
      CL          = CL  +DZ*(SCL(K+1)*CHORD(K+1)  +SCL(K)*CHORD(K))
      CD          = CD  +DZ*(SCD(K+1)*CHORD(K+1)  +SCD(K)*CHORD(K))
      CMP         = CMP  +DZ*(CHORD(K+1)*(SCM(K+1)*CHORD(K+1)
     1                         -SCL(K+1)*XC(K+1))
     2                  +CHORD(K)*(SCM(K)*CHORD(K)
     3                         -SCL(K)*XC(K)))
      CMR         = CMR  +AZ*DZ*(SCL(K+1)*CHORD(K+1)  +SCL(K)*CHORD(K))
      CMY         = CMY  +AZ*DZ*(SCD(K+1)*CHORD(K+1)  +SCD(K)*CHORD(K))
   12 S           = S  +DZ*(CHORD(K+1)  +CHORD(K))
      CL          = CL/S
      CD          = CD/S
      CMP         = CMP*SPAN/S**2
      CMR         = (CMR  +CMR)/(S*SPAN)
      CMY         = (CMY  +CMY)/(S*SPAN)
      RETURN
      END




*DECK PSURE
      SUBROUTINE PSURE(IPLOT,K,X,Y,CP,I1,I2,CL,CD,CM)
C     GENERATES PLOT FOR PRESSURE DISTRIBUTION OVER BLADE SECTION
C     AT EQUAL INTERVALS IN THE MAPPED PLANE
*CALL BLANK
      DIMENSION R(100),D1(150),D2(150),D3(150)
      DIMENSION X(1),Y(1),CP(1)
      IF (IPLOT) 1,11,101
    1 CALL VERSA(20)
      CALL BGNPL(-1)
      IPLOT       = 0
   11 CALL PHYSOR(0.,0.)
      CALL TITLE(1H ,0,1H ,0,1H ,0,8.,10.5)
      CALL GRAPH(0.,1.,0.,1.)
      ZSO         = (CO(K) +HINGE)/(CO(KTE2)+HINGE)
      ZS          = CO(K) + HINGE
      VROT        = OMEGA*ZS
      VTAN        = VROT +FMACH*SIN(PSI)*COS(ALPHA)
      SMACH(K)    = VTAN
      T1          = 1./(.7*SMACH(K)**2)
      PSID        = PSI*RAD
```

```
      ENCODE(45,4,R) PSID,FMACH,TMACH
    4 FORMAT(6HPSI   =,F7.1,3X,6HFMACH=,F7.4,3X,6HTMACH=,F7.4)
      CALL MESSAG(R,45,1.5,1.)
      ENCODE(45,15,R) ZS0,SMACH(K),TILT
   15 FORMAT(6HZS    =,F7.4,3X,6HSMACH=,F7.4,3X,6HAL    =,F7.4)
      CALL MESSAG(R,45,1.5,0.75)
      ENCODE(45,16,R) CL,CD,CM
   16 FORMAT(6HCL    =,F7.4,3X,6HCD    =,F7.4,3X,6HCM    =,F7.4)
      CALL MESSAG(R,45,1.5,.5)
      ENCODE(2,17,R)
   17 FORMAT(2HCP)
      CALL MESSAG(R,2,1.4,5.25)
C     DRAW AIRFOIL
      XMAX       = X(I1)
      XMIN       = X(I1)
      DO 22 I =I1,I2
      XMAX       = AMAX1(X(I),XMAX)
   22 XMIN       = AMIN1(X(I),XMIN)
      SCALE      = 5./(XMAX  -XMIN)
      XOR        = 2.
      YOR        = 2.
      N          = I2-I1+1
      DO 24 J=1,N
      D1(J)      = SCALE*(X(J+I1-1)-XMIN) +XOR
   24 D2(J)      = SCALE*Y(J+I1-1) +YOR
      CALL CURVE(D1,D2,N,0)
      CPMAX      = 0.
      IMAX       = I1
      DO 25 I= I1,I2
      ABSCP      = CP(I)
      IF(ABSCP.LE.CPMAX) GO TO 25
      CPMAX      = ABSCP
      IMAX       = I
   25 CONTINUE
      YOR        = YOR + 3.
C     CPC IS CRITICAL PRESSURE COEFFICIENT
      AAC        = (1.+.2*SMACH(K)**2)/1.2
      CPC        = (AAC**3.5 - 1. )*T1
      IF(ABS(CPC).GT.1.2) GO TO 50
      CPCM       = YOR-2.5*CPC
      CALL STRTPT(2.,CPCM)
      CALL CONNPT(3.,CPCM)
   50 N          = IMAX -I1 +1
      DO 32 J=1,N
      D3(J)      = D1(J)
   32 D2(J)      = YOR -2.5*CP(J+I1-1)
      CALL MARKER(4)
      CALL CURVE(D3,D2,N,0)
      N          = I2 - IMAX + 1
      DO 34 J= 1,N
      D3(J)      = D1(J+IMAX-I1)
   34 D2(J)      = YOR-2.5*CP(J+IMAX-1)
      CALL MARKER(3)
```

```
      CALL CURVE(D3,D2,N,0)
      CALL ENDGR(0)
C     DRAW CP AXIS
      CALL OREL(2.,2.)
      CALL TITLE(1H ,0,1H ,0,1H ,1,6.,6.)
      CALL YAXANG(0.)
      CALL GRAPH(0.,1.,1.2,-.4)
      CALL ENDPL(0)
  101 RETURN   .
      END
```

```
*DECK THREED
      SUBROUTINE THREED(IPLOT,SU,SV,SW,SM,CP,X,Y,TYTLE,CHORD,
     1                  CL,CD,CHORDO,SCL,SCD,SCM)
C     GENERATES PLOT FOR PRESURE DISTRIBUTIONS OVER BLADE
*CALL BLANK
*CALL A
      DIMENSION    X(1),Y(1),SU(1),SV(1),SW(1),SM(1),CP(1),
     .             SCL(1),SCD(1),SCM(1),CHORD(1),TYTLE(1),
     .             XD(200),YD(200),CPD(200),R(80)
      IF (IPLOT)1,11,101
    1 CALL VERSA(20)
      CALL BGNPL(-1)
      IPLOT      = 0
   11 CALL PHYSOR(0.,0.)
      CALL TITLE(1H ,0,1H ,0,1H ,0,8.,10.5)
      CALL GRAPH(0.,1.,0.,1.)
      SPAN       = CO(KTE2)  -CO(KTE1)
      AR         = SPAN/CHORDO
      SCALXX     = 2.5/CHORDO
      SCALZZ     = 5./SPAN
      SCALPP     = -1.25
      TX         = 3.5
      XOR        = 4.5 -SCALXX *XC(KTE1)
      YOR        = 3.75
      DO 6 K= KTE1,KTE2
      I1         = ITE1(K)
      I2         = ITE2(K)
      CALL VELO(K,SU,SV,SW,SM,CP,X,Y)
      CHORD(K)   = X(I1)  -X(LX)
      CALL FORCF (I1,I2,X,Y,CP,TILT,CHORD(K),XC(K),SCL(K),SCD(K),SCM(K))
      SY         = SCALZZ*(CO(K)  -CO(KTE1)) +YOR
      DO 7 I= I1,LX
      J=I-I1+1
      XD(J)      = SCALXX*X(I) +XOR
    7 CPD(J)     = SCALPP*CP(I)  +SY
      N          = LX  -I1  +1
      CALL CURVE(XD,CPD,N,0)
      DO  8 I=LX,I2
      J          = I-LX+1
```

```
      XD(J)       = SCALXX*X(I) +XOR -TX
    8 CPD(J)      = SCALPP*CP(I)  +SY
      N           = I2  -LX  +1
      CALL CURVE(XD,CPD,N,0)
    6 CONTINUE
      CALL MESSAG(49HUPPER SURFACE PRESSURE     LOWER SURFACE PRESSURE,
     .            49,1.5,1.5)
      CALL MESSAG(TYTLE,100,1.,1.)
      ENCODE(45,3,R) FMACH,TMACH,TILT
    3 FORMAT(6HFMACH=,F7.4,3X,6HTMACH=,F7.4,3X,6HALPHA=,F7.4)
      CALL MESSAG(R,45,1.,0.75)
      CALL TOTFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,CO,XC,
     1            CL,CD1,CMP,CMR,CMY)
      CD          = CD1
      PSID        = PSI*RAD
      ENCODE(45,4,R) PSID,CL,CD
    4 FORMAT(6HPSI   =,F7.1,3X,6HCL    =,F7.4,3X,6HCD    =,F7.4)
      CALL MESSAG(R,45,1.,0.5)
      CALL ENDPL(0)
  101 RETURN
      END
```

```
*DECK ROTORB
      SUBROUTINE ROTORB(IPLOT,SU,SV,SW,SM,CP,X,Y)
C     GENERATES PLOT FOR ROTOR BLADE GEOMETRY
*CALL BLANK
*CALL A
      DIMENSION   X(1),Y(1),SU(1),SV(1),SW(1),SM(1),CP(1),
     .            D1(200),D2(200),D3(200),D4(200),D5(200),
     .            XSMAX(50),XSMIN(50),ZSTAT(50),R(80)
      IF (IPLOT)1,11,101
    1 CALL VERSA(20)
      CALL BGNPL(-1)
      IPLOT       = 0
   11 CALL PHYSOR(0.,0.)
      CALL TITLE(1H ,0,1H ,0,1H ,0,8.,10.5)
      CALL GRAPH(0.,1.,0.,1.)
      SPAN        = CO(KTE2)  -CO(KTE1)
      SCALZZ      = 7./SPAN
      CALL MESSAG(21HONERA SWEPT TIP BLADE,21,2.,1.)
      DO 6 K= KTE1,KTE2
      I1          = ITE1(K)
      I2          = ITE2(K)
      CALL VELO(K,SU,SV,SW,SM,CP,X,Y)
      XMAX        = -10.
      XMIN        = 10.
      DO 7 I=I1,I2
      XMAX        = AMAX1(XMAX,X(I))
    7 XMIN        = AMIN1(XMIN,X(I))
      IF(K.EQ.KTE1) XSTAT= XMIN
```

```
      XSMAX(K)   = SCALZZ*(XMAX -XSTAT)
      XSMIN(K)   = SCALZZ*(XMIN -XSTAT)
      ZS         = CO(K) -CO(KTE1)
      ZSTAT(K)   = SCALZZ*ZS
    6 CONTINUE
      DO 8 K=KTE1,KTE2
      KK         = K-KTE1+1
      D1(KK)     = 2.+XSMAX(K)
      D2(KK)     = 2.+XSMIN(K)
    8 D3(KK)     = 2.+ZSTAT(K)
      N          = KTE2-KTE1+1
      CALL CURVE(D1,D3,N,0)
      CALL CURVE(D2,D3,N,0)
C     DRAW AIRFOIL
      DO 21 KK=KTE1,KTE2
      K          = KK-KTE1 +1
      I1         = ITE1(KK)
      I2         = ITE2(KK)
      CALL VELO(KK,SU,SV,SW,SM,CP,X,Y)
      N          = I2-I1+1
      DO 24 J=1,N
      D4(J)      = SCALZZ*(X(J+I1-1)-XSTAT) +2.
   24 D5(J)      = SCALZZ*Y(J+I1-1) +D3(K)
      CALL CURVE(D4,D5,N,0)
   21 CONTINUE
      CALL ENDPL(0)
  101 RETURN
      END
```

```
*DECK VELO
      SUBROUTINE VELO(K,SU,SV,SW,SM,CP,X,Y)
C     CALCULATES SURFACE VELOCITY
C     CP SCALED BY FAR FIELD SOUND SPEED
*CALL BLANK
*CALL A
      DIMENSION   SU(1),SV(1),SW(1),SM(1),CP(1),X(1),Y(1)
      AAO        = 1.
      I1         = ITE1(K)
      I2         = ITE2(K)
      ZS         = CO(K) + HINGE
      VROT       = OMEGA*ZS
      VTAN       = VROT +FMACH*SIN(PSI)*COS(ALPHA)
      SMACH(K)   = VTAN
      T1         = 1./(.7*SMACH(K)**2)
      DO 12 I=I1,I2
      X1         = AO(I)
      Y1         = SO(I,K)
      X1X1       = X1*X1
      Y1Y1       = Y1 *Y1
      HH         = X1X1 + Y1Y1
```

```
      DHH          = 1./ HH
      XB           = .5*(X1X1 -Y1Y1)
      YB           = X1*Y1
      XS           = XC(K) +XB*SCAL
      X1XB         = X1 *DHH
      X1YB         = Y1* DHH
      X1ZB         = -XZ(K) *X1XB -YZ(K) *X1YB
      Y1ZB         = XZ(K) *X1YB - YZ(K) *X1XB
      YXB          = -(X1YB +X1XB*SX(I,K))
      YYB          =   X1XB -X1YB*SX(I,K)
      YZB          =   Y1ZB -X1ZB*SX(I,K) -SZ(I,K)
      GI           = G(I+1,2,K) -G(I-1,2,K)
      GJ           = 2.*(G(I,3,K)-G(I,2,K))
      GK           = G(I,2,K+1)  -G(I,2,K-1)
      GX           = A1(1)*GI
      GY           = B1(2)*GJ
      GZ           = C1(K)*GK
      U            = (GX*X1XB+GY*YXB)/SCAL
      V            = (GX*X1YB+GY*YYB)/SCAL
      W            = (GX*X1ZB+GY*YZB+GZ)/SCAL
      QQ           = U*U+ V*V + W*W
      UF           = OMEGA*ZS +CAC
      VF           = SA
      WF           = -(OMEGA*XS +CAS)
      TERMS        = U*UF +V*VF +W*WF
      FIT          = TERMS
      AA           = DIM(AA0,.2*QQ+.4*FIT)
      UB           = U +UF
      VB           = V +VF
      WB           = W +WF
      UUB          = UB*UB
      VVB          = VB*VB
      WWB          = WB*WB
      QQR          = UUB +VVB +WWB
      SU(I)        = UB
      SV(I)        = VB
      SW(I)        = WB
      SM(I)        = SQRT(QQR/AA)
      CP(I)        = (AA**3.5-1.)*T1
      X(I)         = XS
   12 Y(I)         = YC(K)  +SCAL*YB
      RETURN
      END
```

```
*DECK ESTIM
      SUBROUTINE ESTIM
C     INITIALIZATION FOR STEADY CALCULATION
*CALL BLANK
*CALL A
      LX           = NX/2  +1
```

```
      DX          = 2./FLOAT(NX)
      DSUM        = 1./(A2(LX) +B2(2))
      WATY        = B2(2)*DSUM
      WATX        = A2(LX)*DSUM
      AA0         = 1.
      MX          = NX   +1
      MY          = NY   +2
      MZ          = NZ   +1
      DO 17 J= 1,MY
      DO 17 K= 1,MZ
      DO 17 I= 1,MX
      G(I,J,K)   = 0.
   17 CONTINUE
C     SURFACE CONDITION
      DO 23 K=2,NZ
      IF(ITE2(K).EQ.MX) GO TO 23
      ZS          = CO(K) +HINGE
      IX1         = ITE1(K)
      IX2         = ITE2(K)
      DO 22 I=IX1,IX2
      X1          = A0(I)
      Y1          = S0(I,K)
      X1X1        = X1 *X1
      Y1Y1        = Y1 *Y1
      HH          = X1X1 +Y1Y1
      DHH         = 1./HH
      XB          = .5*(X1X1 -Y1Y1)
      XS          = XC(K) +XB*SCAL
      X1XB        = X1 *DHH
      X1YB        = Y1 *DHH
      X1ZB        = -XZ(K) *X1XB -YZ(K) *X1YB
      Y1ZB        =  XZ(K) *X1YB -YZ(K) *X1XB
      GI          = G(I+1,2,K) -G(I-1,2,K)
      GK          = G(I,2,K+1) -G(I,2,K-1)
      GX          = A1(I)*GI
      GZ          = C1(K)*GK
      UF          = OMEGA*ZS +CAC
      VF          = SA
      WF          = -(OMEGA*XS +CAS)
      YXB         = -(X1YB +X1XB*SX(I,K))
      YYB         =   X1XB -X1YB*SX(I,K)
      YZB         = Y1ZB -X1ZB*SX(I,K) -SZ(I,K)
      X1YS        = X1XB*YXB +X1YB*YYB +X1ZB*YZB
      YYS         = YXB*YXB +YYB*YYB +YZB*YZB
      RHS         = (UF*YXB +VF*YYB +WF*YZB)*SCAL
      G(I,1,K)    = G(I,3,K) +(RHS +X1YS*GX +YZB*GZ)/(YYS*B1(2))
   22 CONTINUE
   23 CONTINUE
      RETURN
      END
```

```
*DECK REFIN
      SUBROUTINE REFIN
*CALL BLANK
*CALL A
      LX             = NX/2   +1
      AAO            = 1.
      DSUM           = 1./(A2(LX) +B2(2))
      WATY           = B2(2)*DSUM
      WATX           = A2(LX)*DSUM
      DX             = 2./NX
      MX             = NX   +1
      MY             = NY   +2
      MZ             = NZ   +1
      MXO            = NX/2   +1
      MYO            = NY/2   +2
      MZO            = NZ/2   +1
      DO 1 MK=1,MZO
      K              =MZO   +1   -MK
      KK             = (K-1)*2   +1
      DO 1 MJ=2,MYO
      J              =MYO   +2   -MJ
      JJ             = (J-2)*2   +2
      DO 1 MI=1,MXO
      I              = MXO   +1   -MI
      II             = (I-1)*2   +1
    1 G(II,JJ,KK)=G(I,J,K)
      DO 2 K=1,MZ,2
      DO 3 J=2,MY,2
      DO 3 I=2,MX,2
    3 G(I,J,K)   =  .5*(G(I+1,J,K)   +G(I-1,J,K))
      DO 4 I=1,MX
      DO 4 J=3,MY,2
    4 G(I,J,K)   =  .5*(G(I,J+1,K)   +G(I,J-1,K))
    2 CONTINUE
      DO 5 K=2,MZ,2
      DO 5 J=2,MY
      DO 5 I=1,MX
    5 G(I,J,K)   =  .5*(G(I,J,K+1)   +G(I,J,K-1))
      DO 6 K=2,NZ
      IX1            = ITE1(K)
      IX2            = ITE2(K)
      IF(IX2.EQ.MX) GO TO 7
      ZS             = CO(K) +HINGE
C     WING CONDITION
      DO 10 I=IX1,IX2
      X1             = AO(I)
      Y1             = SO(I,K)
      X1X1           = X1 *X1
      Y1Y1           = Y1 *Y1
      HH             = X1X1 +Y1Y1
      DHH            = 1. /HH
      XB             = .5*(X1X1 -Y1Y1)
      XS             = XC(K) +XB*SCAL
      X1XB           = X1 *DHH
      X1YB           = Y1 *DHH
```

```
      X1ZB       = -XZ(K) *X1XB -YZ(K) *X1YB
      Y1ZB       =  XZ(K) *X1YB -YZ(K) *X1XB
      GI         = G(I+1,2,K) -G(I-1,2,K)
      GK         = G(I,2,K+1) -G(I,2,K-1)
      GX         = A1(I)*GI
      GZ         = C1(K)*GK
      UF         = OMEGA*ZS +CAC
      VF         = SA
      WF         = -(OMEGA*XS +CAS)
      YXB        = -(X1YB +X1XB*SX(I,K))
      YYB        =   X1XB -X1YB*SX(I,K)
      YZB        = Y1ZB -X1ZB*SX(I,K) -SZ(I,K)
      X1YS       = X1XB*YXB +X1YB*YYB +X1ZB*YZB
      YYS        = YXB*YXB +YYB*YYB +YZB*YZB
      RHS        = (UF*YXB +VF*YYB +WF*YZB)*SCAL
   10 G(I,1,K)   = G(I,3,K) +(RHS +X1YS*GX +YZB*GZ)/(YYS*B1(2))
      E          = G(IX2,2,K)-G(IX1,2,K)
      IX         = IX2 +1
      DO 8 I=IX,MX
      M          = NX  +2  -I
      G(I,1,K)   = G(M,3,K)  +E
    8 G(M,1,K)   = G(I,3,K)  -E
      GO TO 6
    7 G(LX,2,K) = G(LX,3,K)*WATY+G(LX-1,2,K)*WATX
      DO 9 I=LX,MX
      M          = NX  +2  -I
      G(I,2,K)   = G(M,2,K)
      G(I,1,K)   = G(M,3,K)
    9 G(M,1,K)   = G(I,3,K)
    6 CONTINUE
      RETURN
      END
*DECK RELAX
      SUBROUTINE RELAX
*CALL BLANK
*CALL A
*CALL FLO
      DIMENSION    C(131),D(131),GM(129,18,33),
     .             AB(129),AC(129),AA(129),QQR(129),R(129),
     .             HH(129),XX1S(129),YYS(129),X1YS(129),
     .             X1ZB(129),YZB(129),GI(129),GJ(129),GK(129),
     .             GII(129),GJJ(129),GKK(129),
     .             GIJ(129),GIK(129),GJK(129),
     .             UUR(129),VVR(129),WWR(129),
     .             UVR(129),UWR(129),VWR(129),
     .             UR(129),VR(129),WR(129)
      T1         = DX*DX
      Q1         = 2./P1
      Q2         = 1./P2
      FR         = 0.
      IR         = 0
      JR         = 0
      KR         = 0
      GD         = 0.
      IG         = 0
```

```
      JG         = 0
      KG         = 0
      NS         = 0
      C(1)       = 0.
      D(1)       = 0.
      DO 70 K=1,MZ
      DO 70 J=1,MY
      DO 70 I=1,MX
      GM(I,J,K)    = G(I,J,K)
   70 CONTINUE
  303 DO 103 K=2,NZ
      ZS         = CO(K) +HINGE
C     FOR FIXED WING FLOW J=NY+1
      J          = NY
      I3         = NX
   31 BC         = T1*B1(J)*C1(K)
C     INTERIOR
  403 DO 400 I= 2,I3
      AB(I)      = T1*A1(I) *B1(J)
      AC(I)      = T1*A1(I) *C1(K)
      X1         = A0(I)
      Y1         = B0(J) +S0(I,K)
      X1X1       = X1 *X1
      Y1Y1       = Y1 *Y1
      HH(I)      = X1X1 +Y1Y1
      DHH        = 1. /HH(I)
      XB         = .5*(X1X1 -Y1Y1)
      XS         = XC(K) +XB*SCAL
      X1XB       = X1 *DHH
      X1YB       = Y1 *DHH
      X1ZB(I)    = -XZ(K) *X1XB -YZ(K) *X1YB
      Y1ZB       =  XZ(K) *X1YB -YZ(K) *X1XB
      YXB        = -(X1YB +X1XB*SX(I,K))
      YYB        =   X1XB -X1YB*SX(I,K)
      YZB(I)     =   Y1ZB -X1ZB(I)*SX(I,K) -SZ(I,K)
      GI(I)      = G(I+1,J,K) -G(I-1,J,K)
      GJ(I)      = G(I,J+1,K) -G(I,J-1,K)
      GK(I)      = G(I,J,K+1) -G(I,J,K-1)
      GX         = A1(I)*GI(I)
      GY         = B1(J)*GJ(I)
      GZ         = C1(K)*GK(I)
      U          = (GX*X1XB+GY*YXB)/SCAL
      V          = (GX*X1YB+GY*YYB)/SCAL
      W          = (GX*X1ZB(I)+GY*YZB(I)+GZ)/SCAL
      QQ         = U*U +V*V +W*W
      UF         = OMEGA*ZS +CAC
      VF         = SA
      WF         = -(OMEGA*XS +CAS)
      TERMS      = U*UF +V*VF +W*WF
      FIT        = TERMS
      AA(I)      = DIM(AA0,.2*QQ+.4*FIT)
      UB         = U +UF
      VB         = V +VF
      WB         = W +WF
      UUB        = UB*UB
```

```
      VVB        = VB*VB
      WWB        = WB*WB
      UVB        = UB*VB
      UWB        = UB*WB
      VWB        = VB*WB
      UR(I)      = X1XB*UB +X1YB*VB +X1ZB(I)*WB
      VR(I)      =  YXB*UB +YYB*VB  +YZB(I)*WB
      WR(I)      = WB
      UUR(I)     = UR(I)*UR(I)
      VVR(I)     = VR(I)*VR(I)
      WWR(I)     = WWB
      UVR(I)     = UR(I)*VR(I)
      UWR(I)     = UR(I)*WR(I)
      VWR(I)     = VR(I)*WR(I)
      QQR(I)     = UUB +VVB +WWB
      XX1S(I)    = X1XB*X1XB + X1YB*X1YB + X1ZB(I)*X1ZB(I)
      X1YS(I)    = X1XB*YXB +X1YB*YYB +X1ZB(I)*YZB(I)
      YYS(I)     = YXB*YXB +YYB*YYB +YZB(I)*YZB(I)
      X1XBXB     = -X1*(HH(I) -4.*Y1Y1)*DHH**3
      X1XBYB     =  Y1*(HH(I) -4.*X1X1)*DHH**3
      BCHI       = XZ(K)*X1XBXB +YZ(K)*X1XBYB
      BPSI       = XZ(K)*X1XBYB -YZ(K)*X1XBXB
      BLAMDA     = SX(I,K)*X1XBXB +X1XBYB
      BSIGMA     = SX(I,K)*X1XBYB -X1XBXB
      FA         = XZ(K)*BCHI +YZ(K)*BPSI
      FB         = YZ(K)*BCHI -XZ(K)*BPSI
      FC         = XZ(K)*BLAMDA +YZ(K)*BSIGMA
      FD         = XZ(K)*BSIGMA -YZ(K)*BLAMDA
      FE         = XZZ(K)*X1XB +YZZ(K)*X1YB
      FF         = XZZ(K)*YXB  +YZZ(K)*YYB
      FAA        = FA-FE*SCAL
      FBB        = OMEGA*(CAS-WF)*SCAL
      FCC        = OMEGA*(CAC+UF)*SCAL
      FDD        = FB-FA*SX(I,K)
      FEE        = FDD-FF*SCAL
      RL         = T1*(-FAA*(WWB-AA(I)) -X1XBXB*(UUB-VVB)
     1             +2.*(BCHI*UWB -X1XBYB*UVB +BPSI*VWB)
     2             +FBB*X1XB +FCC*X1ZB(I))
      RM         = T1*(-FEE*(WWB-AA(I))
     1             +BLAMDA*(UUB-VVB)+2.*(UVB*BSIGMA-UWB*FC-VWB*FD)
     2             +FBB*YXB +FCC*YZB(I))
     2             -AA(I)*(XX1S(I)*SXX(1,K) +SZZ(I,K)
     3             +2.*X1ZB(I)*SXZ(1,K))
     4             +UUR(I)*SXX(I,K) +WWB*SZZ(I,K)
     5             +2.*UR(I)*WB*SXZ(I,K)
      RN         = T1*FCC
  400 R(I)       = RL*GX +RM*GY +RN*GZ
      DO 401 I= 2,I3
      GII(I)     = G(I+1,J,K)-2.*G(I,J,K)+G(I-1,J,K) +A3(I)*GI(I)
      GJJ(I)     = G(I,J+1,K)-2.*G(I,J,K)+G(I,J-1,K) +B3(J)*GJ(I)
      GKK(I)     = G(I,J,K+1)-2.*G(I,J,K)+G(I,J,K-1) +C3(K)*GK(I)
      GIJ(I)     = G(I+1,J+1,K)-G(I+1,J-1,K)-G(I-1,J+1,K)+G(I-1,J-1,K)
      GIK(I)     = G(I+1,J,K+1)-G(I+1,J,K-1)-G(I-1,J,K+1)+G(I-1,J,K-1)
      GJK(I)     = G(I,J+1,K+1)-G(I,J+1,K-1)-G(I,J-1,K+1)+G(I,J-1,K-1)
  401 CONTINUE
```

```
      DO 8 I=2,I3
      SIGNX      = SIGN(1.,UR(I))
      SIGNY      = SIGN(1.,VR(I))
      SIGNZ      = SIGN(1.,WR(I))
      AXT        = BETA*UR(I)*A1(I)
      AYT        = BETA*VR(I)*B1(J)
      AZT        = BETA*WR(I)*C1(K)
      LL         = IFIX(SIGNX)
      IM         = I-LL
      IMM        = IM-LL
      LL         = IFIX(SIGNY)
      JM         = J-LL
      JMM        = JM-LL
      LL         = IFIX(SIGNZ)
      KM         = K-LL
      KMM        = KM-LL
      IF(QQR(I).GE.AA(I)) GO TO 9
      AXX        = A2(I)*(UUR(I)-AA(I)*XX1S(I))
      AYY        = B2(J)*(VVR(I)-AA(I)*YYS(I))
      AZZ        = C2(K)*(WWR(I)-AA(I))
      AXY        = 2.*AB(I)*(UVR(I)-AA(I)*X1YS(I))
      AXZ        = 2.*AC(I)*(UWR(I)-AA(I)*X1ZB(I))
      AYZ        = 2.*BC*(VWR(I)-AA(I)*YZB(I))
      YI         = -(AXX*GII(I)+AYY*GJJ(I)+AZZ*GKK(I)
     .             +AXY*GIJ(I)+AXZ*GIK(I)+AYZ*GJK(I))+R(I)
      CI         = -AXX
      BI         = -AXX
      AI         = AXX  +AXX  +Q1*(AYY  +AZZ)
      GO TO 10
C     TYPE DEPENDENT DIFFERENCING
    9 NS         = NS  +1
      BXX        = A2(I) * (QQR(I) *XX1S(I) -UUR(I))
      BYY        = B2(J) * (QQR(I) *YYS(I) -VVR(I))
      BZZ        = C2(K) * (QQR(I) -WWR(I))
      BXY        = 2.*AB(I) * (QQR(I) *X1YS(I) -UVR(I))
      BXZ        = 2.*AC(I) * (QQR(I) *X1ZB(I) -UWR(I))
      BYZ        = 2.*BC * (QQR(I) *YZB(I) -VWR(I))
      DELTA      = BXX*GII(I)+BYY*GJJ(I)+BZZ*GKK(I)
     .             +BXY*GIJ(I)+BXZ*GIK(I)+BYZ*GJK(I)
      IF(IMM.LT.1.OR.IMM.GT.MX) GO TO 11
      GII(I)     = G(I,J,K)-2.*G(IM,J,K)+G(IMM,J,K)
     .             + 2.*A3(I)*SIGNX*(G(I,J,K)-G(IM,J,K))
   11 IF(JMM.LT.1.OR.JMM.GT.MY) GO TO 12
      GJJ(I)     = G(I,J,K)-2.*G(I,JM,K)+G(I,JMM,K)
     .             + 2.*B3(J)*SIGNY*(G(I,J,K)-G(I,JM,K))
   12 IF(KMM.LT.1.OR.KMM.GT.MZ) GO TO 13
      GKK(I)     = G(I,J,K)-2.*G(I,J,KM)+G(I,J,KMM)
     .             + 2.*C3(K)*SIGNZ*(G(I,J,K)-G(I,J,KM))
   13 GIJ(I)     = G(I,J,K)- G(IM,J,K)- G(I,JM,K)+ G(IM,JM,K)
      GIK(I)     = G(I,J,K)- G(IM,J,K)- G(I,J,KM)+ G(IM,J,KM)
      GJK(I)     = G(I,J,K)- G(I,JM,K)- G(I,J,KM)+ G(I,JM,KM)
      AXX        = UUR(I)*A2(I)
      AYY        = VVR(I)*B2(J)
      AZZ        = WWR(I)*C2(K)
      AXY        = 8. *SIGNX *SIGNY *AB(I) *UVR(I)
```

```
      AXZ         = 8. *SIGNX *SIGNZ *AC(I) *UWR(I)
      AYZ         = 8. *SIGNY *SIGNZ *BC *VWR(I)
      GSS         = AXX*GII(I)+AYY*GJJ(I)+AZZ*GKK(I)
     .             +AXY*GIJ(I)+AXZ*GIK(I)+AYZ*GJK(I)
      AQ          = AA(I) / QQR(I)
      YI          = (AQ-1.)*GSS +AQ*DELTA +R(I)
      B           = .5*(AQ  -1.)*(AXX  +AXX  +AXY  +AXZ)
      CI          = AQ*BXX -(1. -SIGNX) *B
      BI          = AQ*BXX -(1. +SIGNX) *B
      AI          = -AQ *(BXX  +BXX  +Q2*(BYY  +BZZ))
     .              +(AQ-1.) *(2. *(AXX +AYY +AZZ) +AXY +AYZ +AXZ)
   10 RES         = ABS(YI)
      IF(RES.LE.FR) GO TO 14
      FR          = RES
      IR          = I
      JR          = J
      KR          = K
   14 IF(SIGNX.GT.0.)GO TO 15
      AI          = AI +AXT
      CI          = CI -AXT
      GO TO 16
   15 AI          = AI -AXT
      BI          = BI +AXT
   16 IF(SIGNY.GT.0.)GO TO 17
      AI          = AI +AYT
      GO TO 18
   17 AI          = AI -AYT
   18 YI          = YI +AYT*SIGNY*(G(I,JM,K)-GM(I,JM,K))
      IF(SIGNZ.GT.0.)GO TO 19
      AI          = AI +AZT
      GO TO 20
   19 AI          = AI -AZT
   20 YI          = YI +AZT*SIGNZ*(G(I,J,KM)-GM(I,J,KM))
      A           = 1./(AI-BI*C(I-1))
      C(I)        = CI*A
    8 D(I)        = (YI-BI*D(I-1))*A
      CG          = 0.
      I           = I3
      DO 42 M=2,I3
      CG          = D(I)  -C(I)*CG
      CORG        = ABS(CG)
      IF (CORG.LE.GD) GO TO 43
      GD          = CORG
      IG          = I
      JG          = J
      KG          = K
   43 G(I,J,K)    = G(I,J,K)-CG
   42 I           = I -1
      J           = J -1
      IF(J-2) 61,51,31
   51 IF (ITE2(K).EQ.MX) I3 = LX -1
      GO TO 31
   61 IF(ITE2(K).EQ.MX) GO TO 113
```

```
      IX1        = ITE1(K)
      IX2        = ITE2(K)
      IX1M       = IX1 -1
      IX2P       = IX2 +1
      E          = G(IX2,2,K) -G(IX1,2,K)
      DO 100 I= 2,IX1M
      M          = NX +2 -I
  100 G(I,1,K)   = G(M,3,K) -E
      DO 62 I=IX1,IX2
      X1         = A0(I)
      Y1         = S0(I,K)
      X1X1       = X1 *X1
      Y1Y1       = Y1 *Y1
      HHS        = X1X1 +Y1Y1
      DHH        = 1. /HHS
      XB         = .5*(X1X1 -Y1Y1)
      XS         = XC(K) +XB*SCAL
      X1XB       = X1 *DHH
      X1YB       = Y1 *DHH
      X1ZBS      = -XZ(K) *X1XB -YZ(K) *X1YB
      Y1ZB       =  XZ(K) *X1YB -YZ(K) *X1XB
      GIS        = G(I+1,2,K) -G(I-1,2,K)
      GKS        = G(I,2,K+1) -G(I,2,K-1)
      GX         = A1(I)*GIS
      GZ         = C1(K)*GKS
      UF         = OMEGA*ZS +CAC
      VF         = SA
      WF         = -(OMEGA*XS +CAS)
      YXB        = -(X1YB +X1XB*SX(I,K))
      YYB        =   X1XB -X1YB*SX(I,K)
      YZBS       = Y1ZB -X1ZBS*SX(I,K) -SZ(I,K)
      X1YSS      = X1XB*YXB +X1YB*YYB +X1ZBS*YZBS
      YYSS       = YXB*YXB +YYB*YYB +YZBS*YZBS
      RHS        = (UF*YXB +VF*YYB +WF*YZBS)*SCAL
   62 G(I,1,K)   = G(I,3,K) +(RHS +X1YSS*GX +YZBS*GZ)/(YYSS*B1(2))
      DO 102 I= IX2P,NX
      M          = NX +2 -I
  102 G(I,1,K)   = G(M,3,K) +E
      GO TO 103
  113 G(LX,2,K)  = G(LX,3,K)*WATY+G(LX-1,2,K)*WATX
      DO 114 I= 2,NX
      M          = NX+2 -I
  114 G(I,1,K)   = G(M,3,K)
      I3         = LX -1
      DO 115 I= 2,I3
      M          = NX+2 -I
  115 G(M,2,K)   = G(I,2,K)
  103 CONTINUE
      RETURN
      END
```

APPENDIX D

LISTING OF SAMPLE DATA

```
ROTOR BLADES
FNX         FNY         FNZ
32.         4.          6.
FIT         COVO        P10         P20         P30         BETA0       FHALF
100.        0.000005    1.75        1.0         1.0         0.1         2.
100.        0.000005    1.75        1.0         1.0         0.1         1.
100.        0.000005    1.75        1.0         1.0         0.1         0.
FSPEED      PSI         ALPHA       TIPWR       RADIUS      AINF
80.         90.         0.0         200.        15.         334.143
CREF        XREF        FBLADE      FCLUST      CD0
.123        .03075      1.          0.          0.
KPLOTS
                                                       010101
FNC         SWEEP1      SWEEP2      SWEEP       DIHED1      DIHED2      DIHED
15.         0.0         0.0         0.0         0.          0.          0.
ZS(K)       XL          YL          CHORD       THICK       TWIST       NEWSEC
1.00        -.25        0.0         1.          1.          0.0         1.0
YSYM        FNU         FNL
1.          65.         65.
X           Y           UPPER SURFACE (NACA 0012)
  0.000000   0.000000
   .000250    .002798
   .000500    .003945
   .000750    .004822
   .001000    .005557
   .001250    .006203
   .001750    .007319
   .003000    .009528
   .003750    .010622
   .004250    .011288
   .005750    .013066
   .006250    .013603
   .007000    .014365
   .008250    .015542
   .009000    .016202
   .009750    .016833
   .011000    .017827
   .011750    .018393
   .012250    .018759
   .022500    .024915
   .030000    .028401
   .037500    .031374
   .050000    .035547
   .057500    .037704
   .062500    .039027
   .072500    .041439
   .077500    .042543
   .082500    .043587
   .092500    .045512
   .097500    .046400
```

```
    .110000    .048432
    .150000    .053452
    .170000    .055287
    .190000    .056760
    .230000    .058790
    .250000    .059412
    .270000    .059807
    .310000    .059994
    .330000    .059820
    .350000    .059486
    .390000    .058387
    .410000    .057643
    .430000    .056780
    .470000    .054732
    .490000    .053560
    .510000    .052298
    .550000    .049524
    .570000    .048021
    .590000    .046447
    .630000    .043096
    .650000    .041325
    .670000    .039494
    .710000    .035659
    .730000    .033658
    .750000    .031603
    .790000    .027332
    .810000    .025117
    .830000    .022848
    .870000    .018148
    .890000    .015715
    .910000    .013225
    .950000    .008066
    .970000    .005393
    .990000    .002654
   1.000000    .001260
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
3.00       -.25       0.0        1.         1.         0.0        0.0
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
5.00       -.25       0.0        1.         1.         0.0        0.0
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
7.00       -.25       0.0        1.         1.         0.0        0.0
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
9.00       -.25       0.0        1.         1.         0.0        0.0
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
11.0       -.25       0.0        1.         1.         0.0        0.0
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
13.0       -.25       0.0        1.         1.         0.0        0.0
ZS(K)      XL         YL         CHORD      THICK      TWIST      NEWSEC
15.0       -.25       0.0        1.         1.         0.0        0.0
%%EOF
```

REFERENCES

1. Caradonna, F. X.; and Isom, M. P.: Subsonic and Transonic Potential Flow over Helicopter Rotor Blades. AIAA J., vol. 10, 1972.

2. Grant, J.: The Prediction of Supercritical Pressure Distributions on Blade Tips of Arbitrary Shape over a Range of Advancing Blade Azimuthal Angles. Fourth European Rotorcraft and Powered Lift Aircraft Forum, 1978.

3. Caradonna, F. X.; and Isom, M.: Numerical Calculations of Unsteady Transonic Potential Flow over Helicopter Rotor Blades. AIAA J., vol. 14, 1976.

4. Chattot, J. J.: Calculation of Three Dimensional Unsteady Transonic Flows Past Helicopter Blades. NASA TP-1721, 1980.

5. Arieli, R.; and Tauber, M.: Computation of Subsonic and Transonic Flow about Lifting Rotor Blades. AIAA Paper 79-1667, 1979.

6. Jameson, A.; and Caughey, D.: Numerical Calculation of the Transonic Flow Past a Swept Wing. Courant Institute of Mathematical Sciences, C003077-140, 1977.

7. Chang, I.; and Tauber, M.: Numerical Calculation of the Transonic Flow Past a Crank Wing. NASA TM-84391, 1983.

8. Bauer, F.; Garabedian, P.; Korn, D.; and Jameson, A.: Supercritical Wing Sections II. Springer Verlag, 1975.

9. Philippe, J. J.; and Chattot, J. J.: Experimental and Theoretical Studies on Helicopter Blade Tips at ONERA. Sixth European Rotorcraft and Powered Lift Aircraft Forum, 1980.

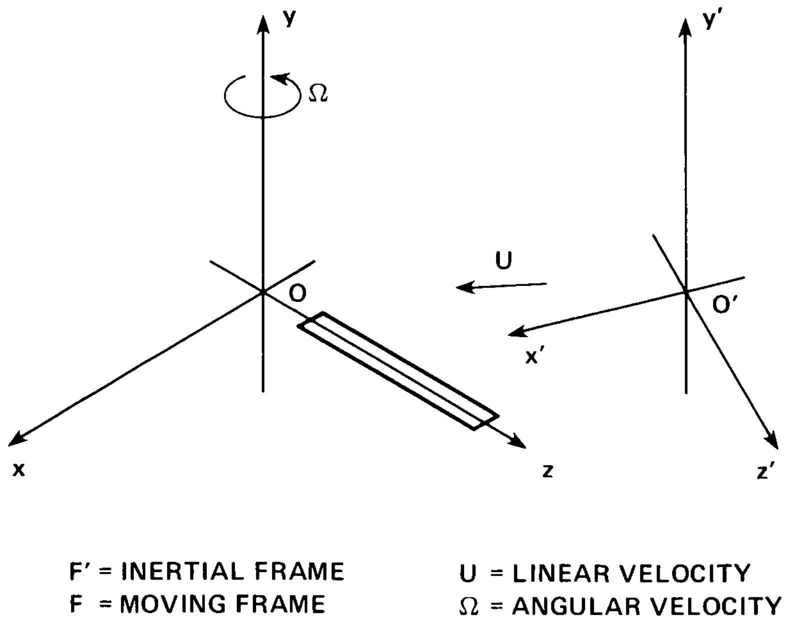


Figure 1.- Rotor coordinate systems.

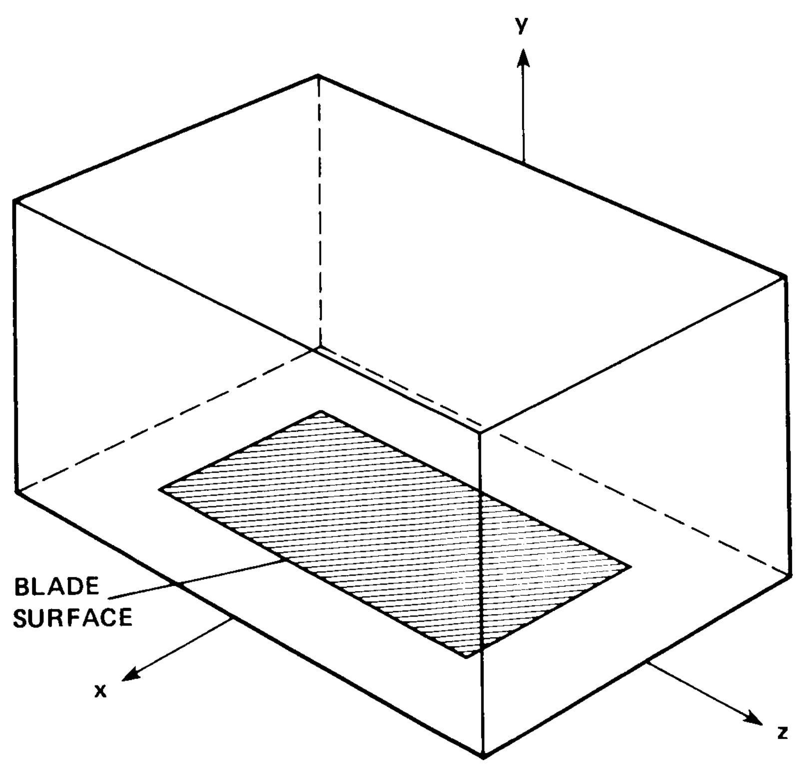
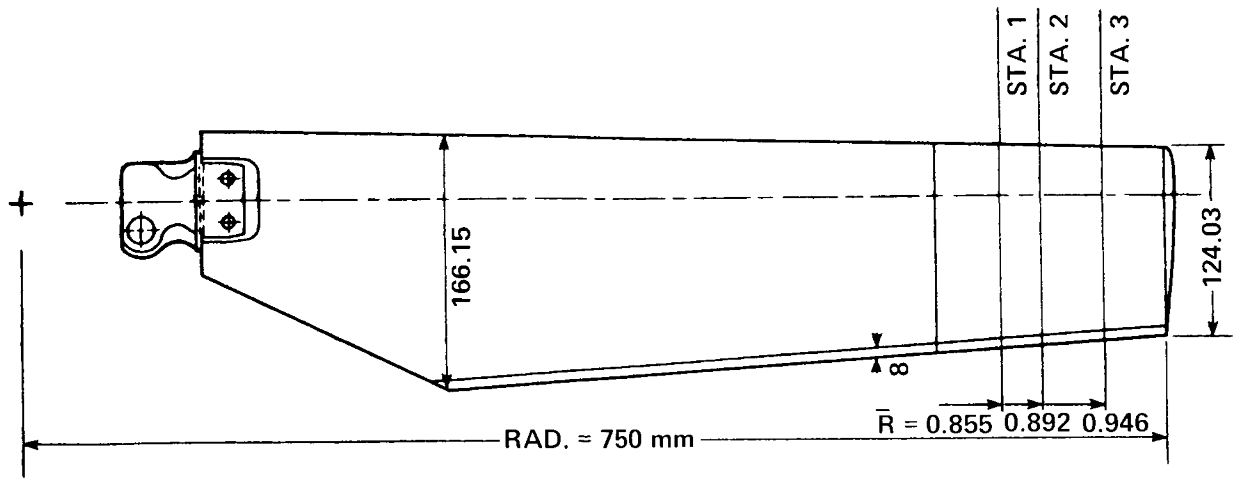
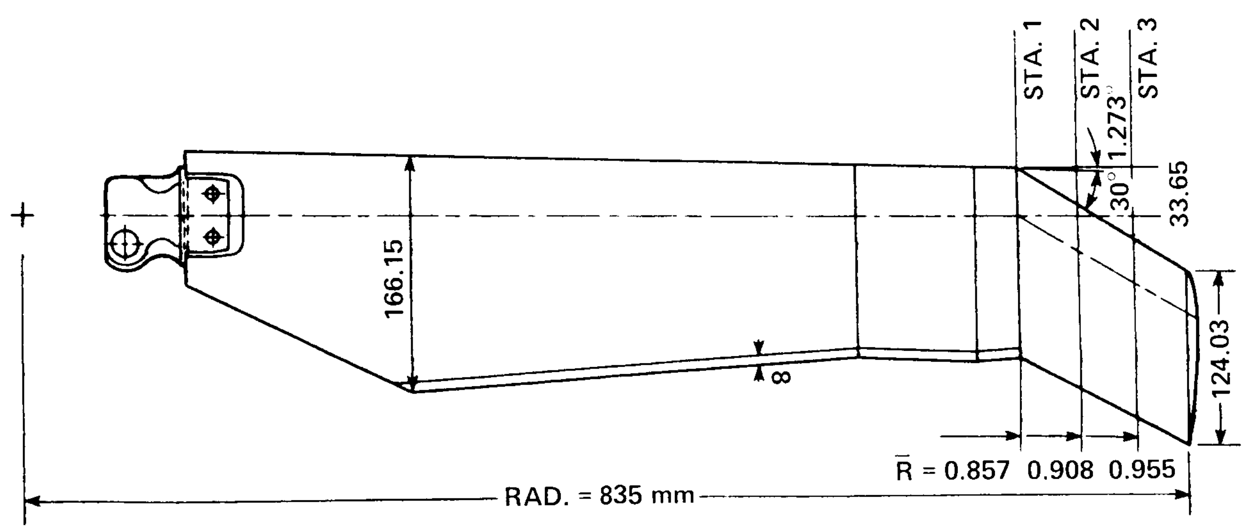


Figure 2.- Sketch of computational domain.

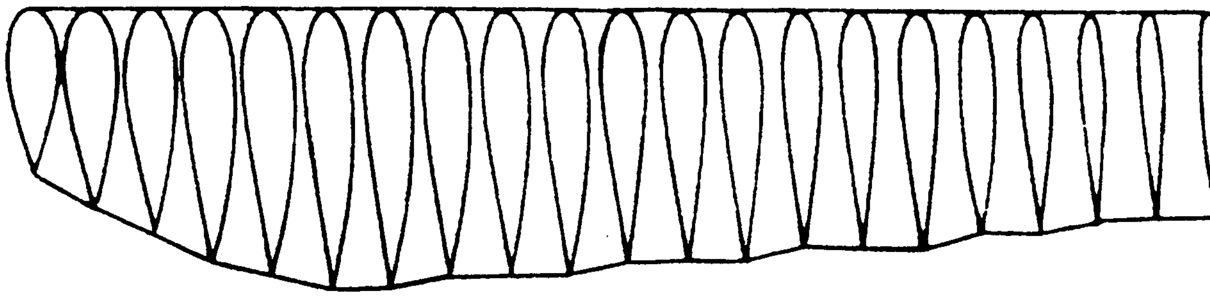


(a) ONERA straight-tip blade geometry.

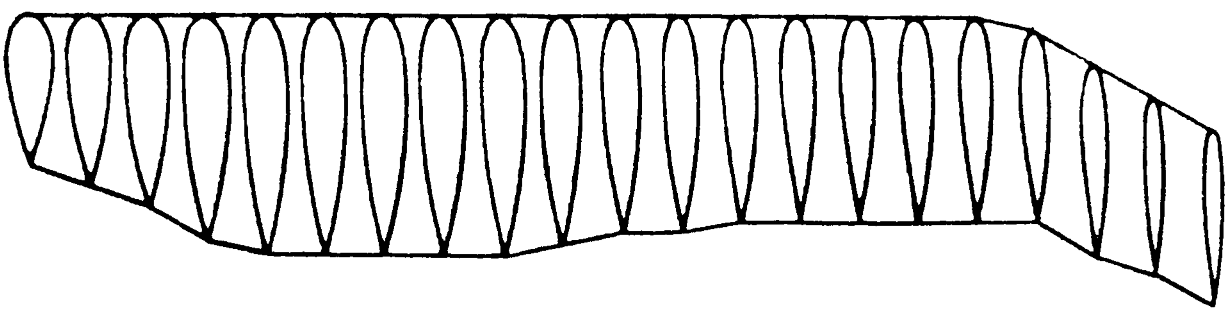


(b) ONERA swept-tip blade geometry.

Figure 3.- ONERA blade geometry.

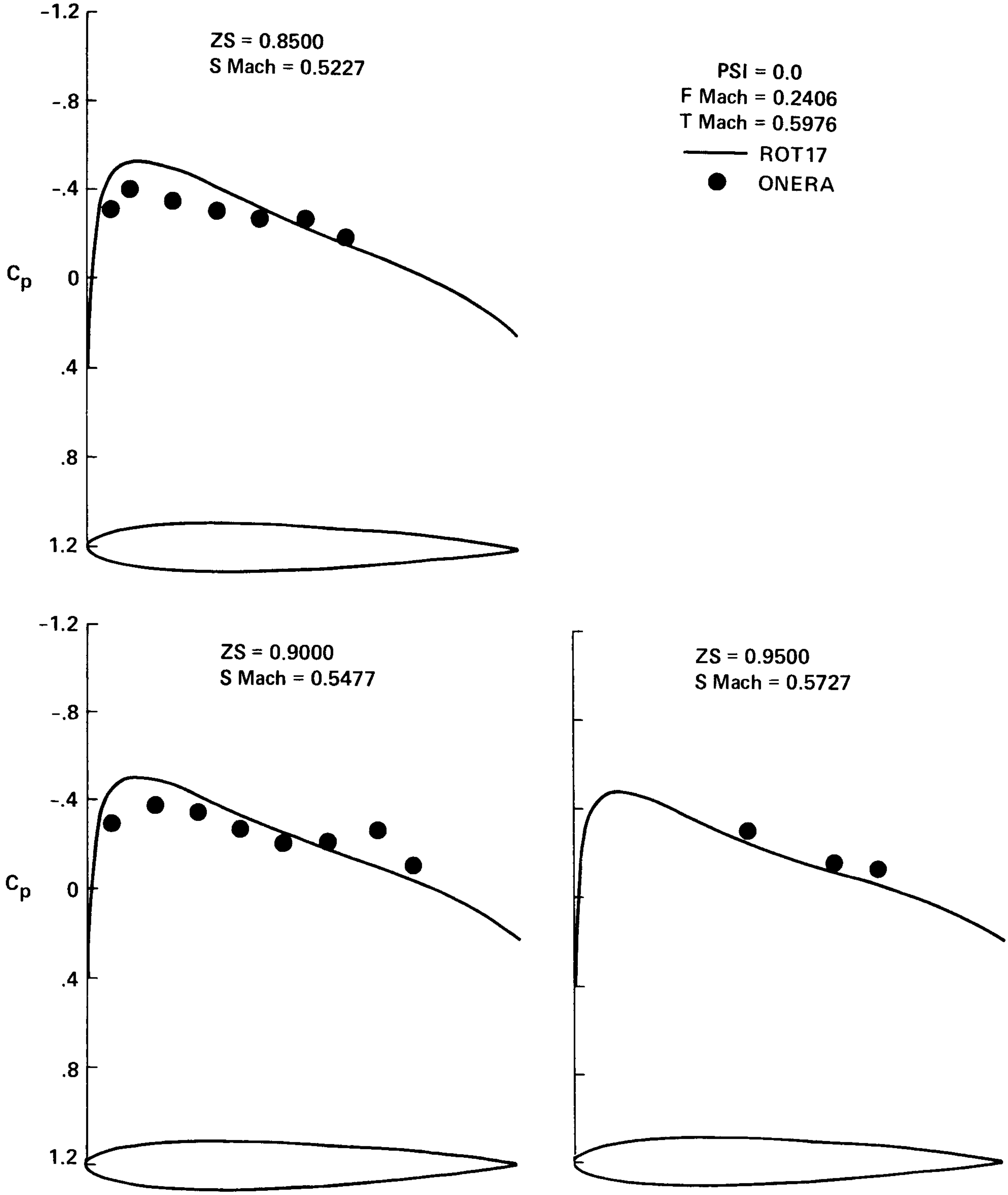


(c) The approximate ONERA straight-tip blade geometry used in the computer code.



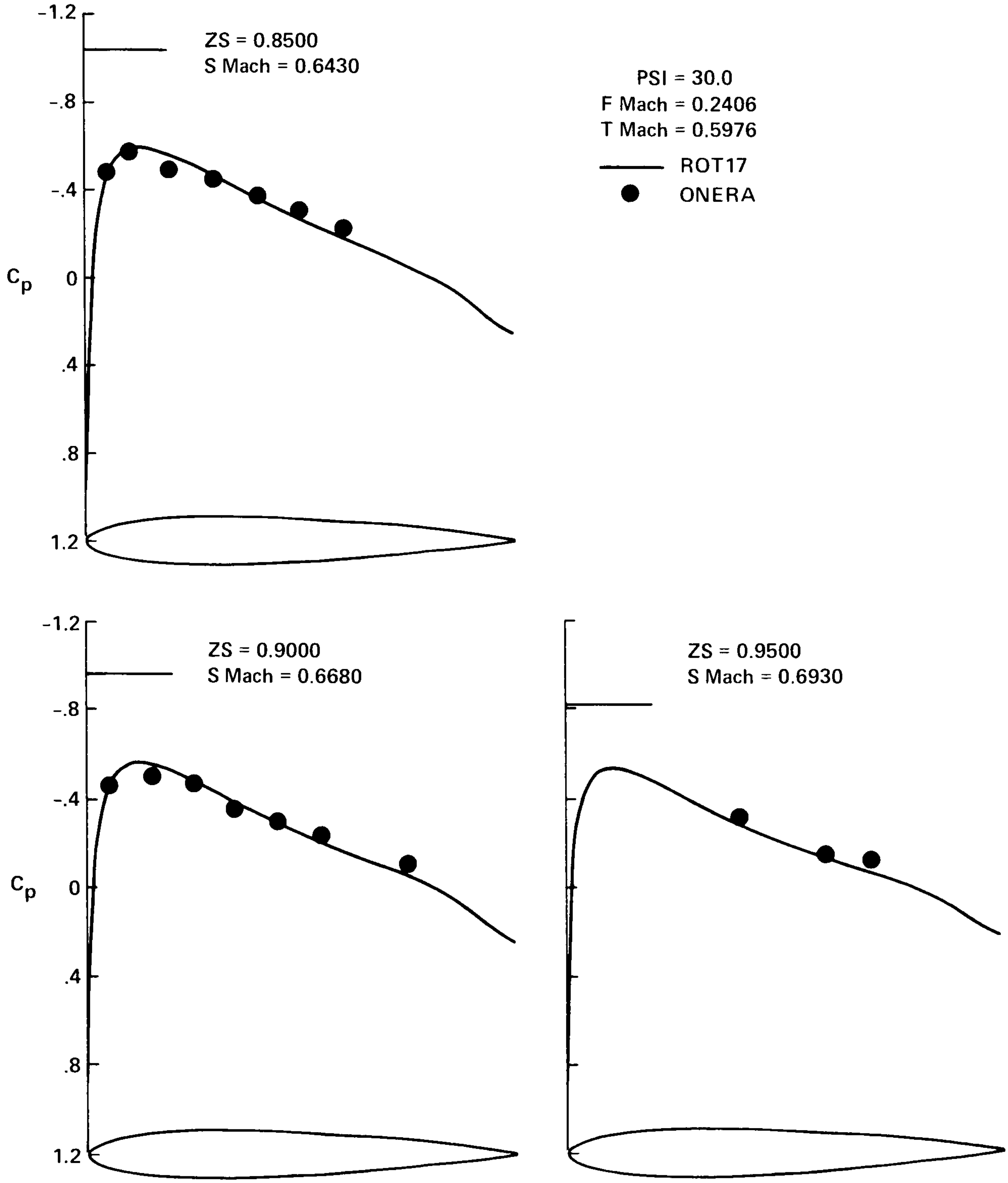
(d) The approximate ONERA swept-tip blade geometry used in the computer code.

Figure 3.- Concluded.



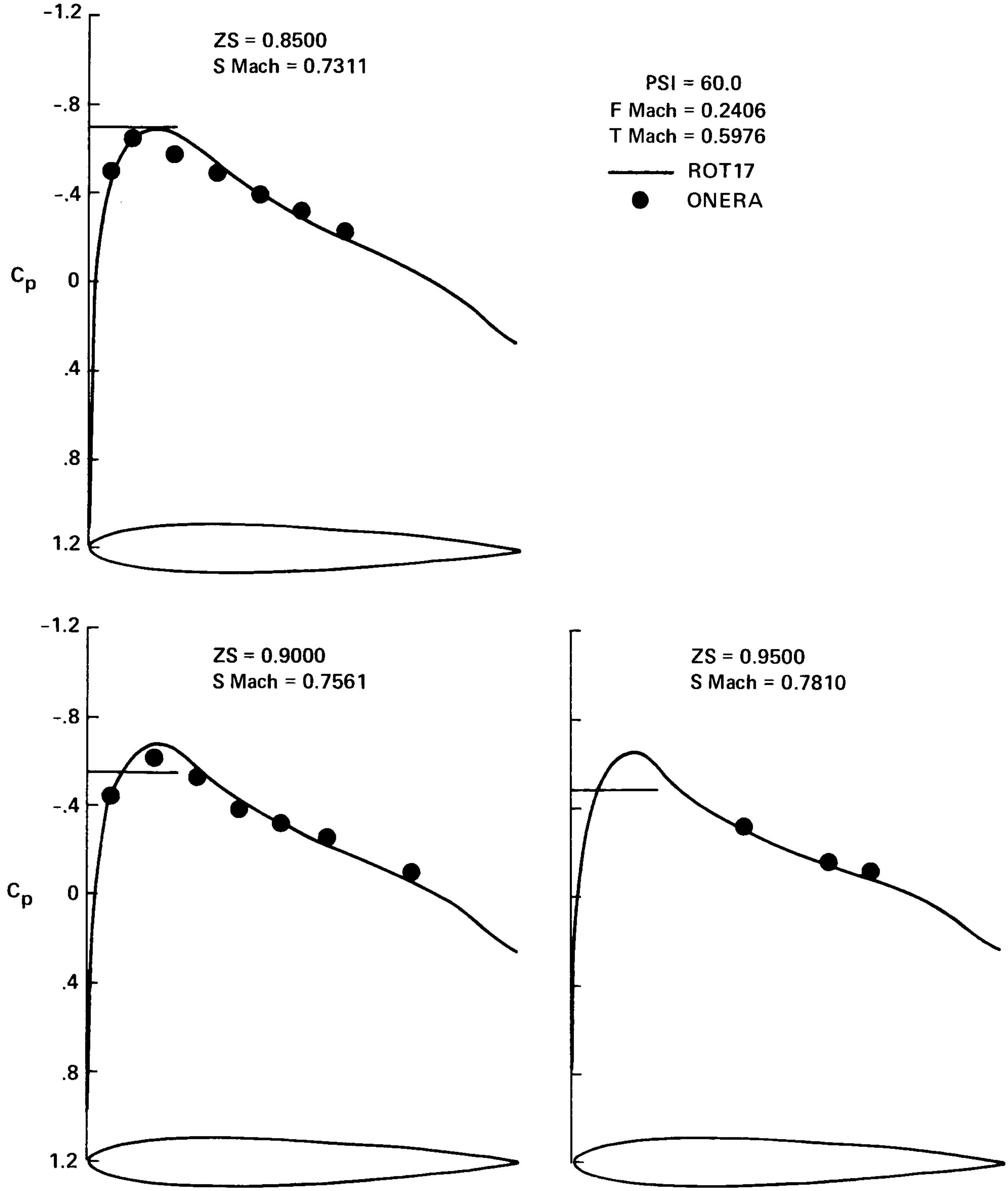
(a) Advance ratio 0.4 at 0° azimuthal angle.

Figure 4.- Comparison between computed and measured surface pressure distributions.



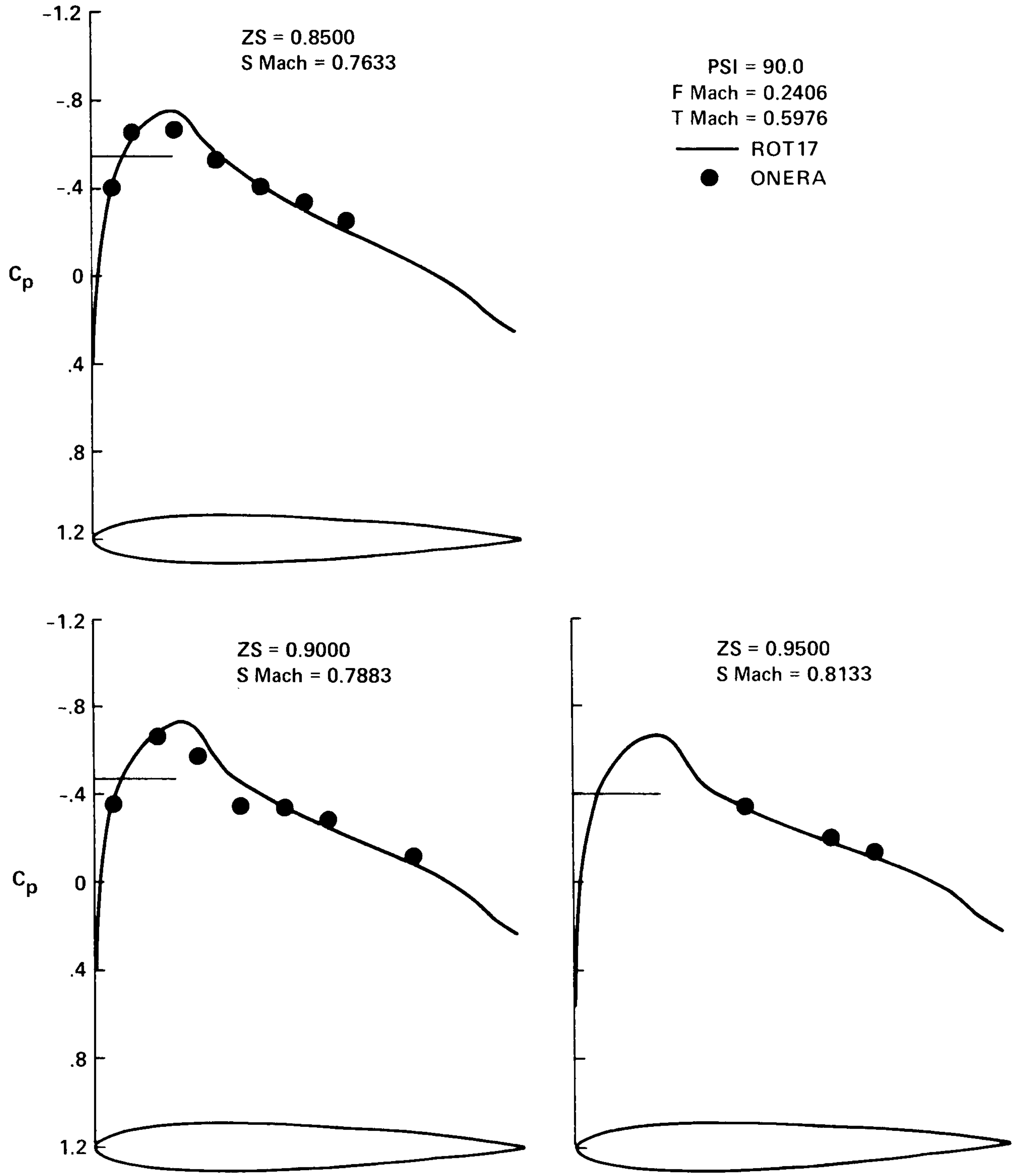
(b) Advance ratio 0.4 at 30° azimuthal angle.

Figure 4.- Continued.



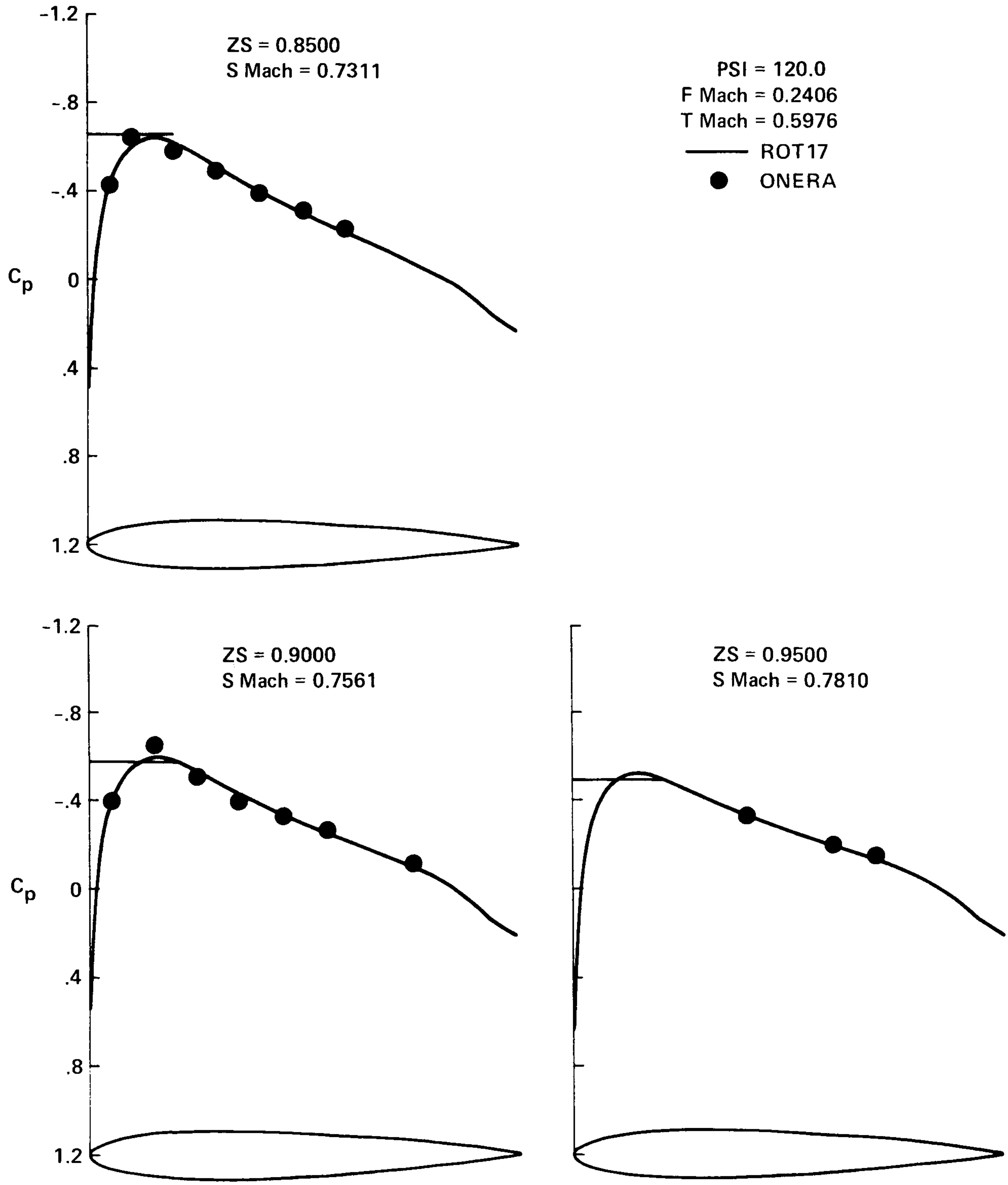
(c) Advance ratio 0.4 at 60° azimuthal angle.

Figure 4.- Continued.



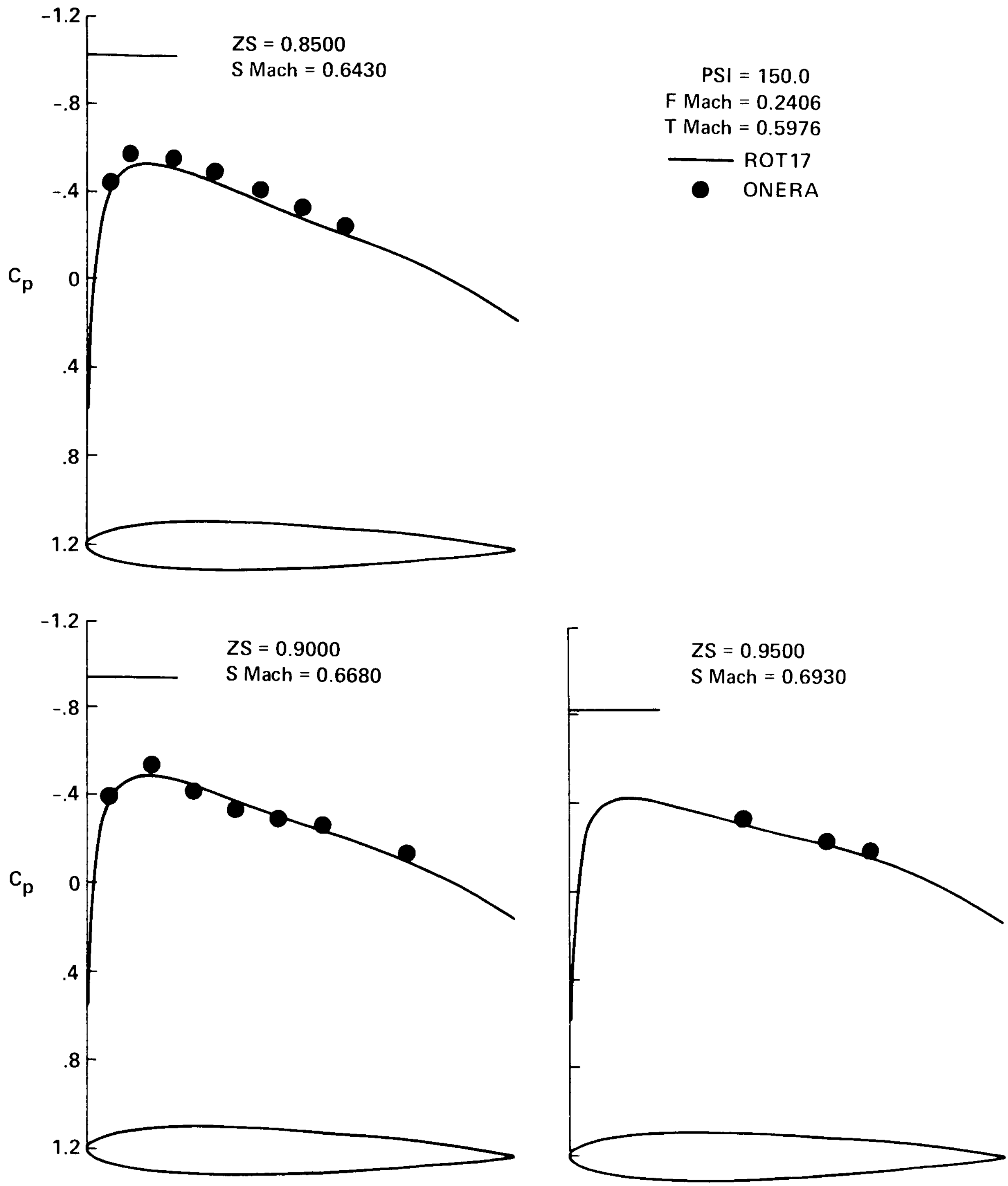
(d) Advance ratio 0.4 at 90° azimuthal angle.

Figure 4.- Continued.



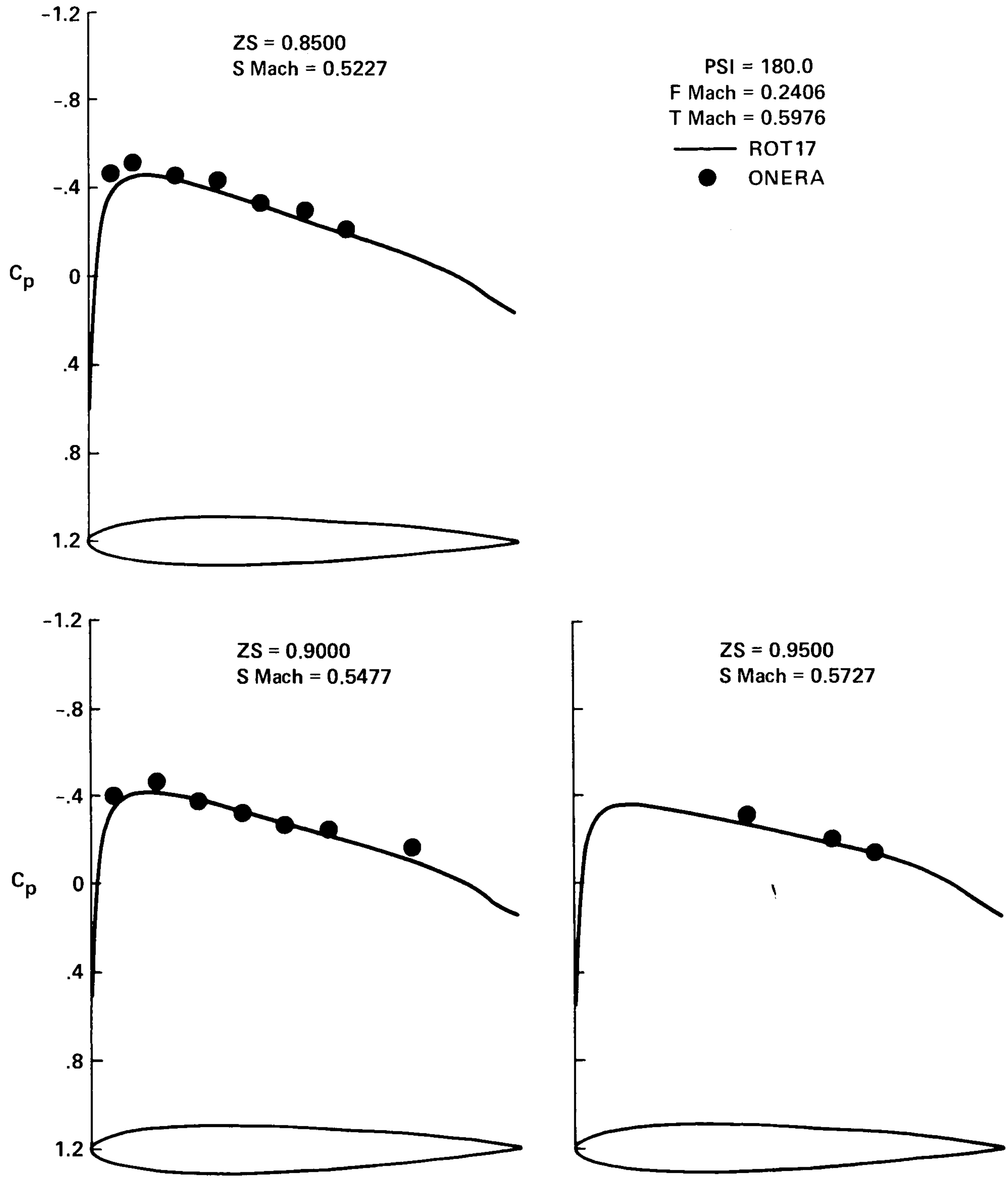
(e) Advance ratio 0.4 at 120° azimuthal angle.

Figure 4.- Continued.



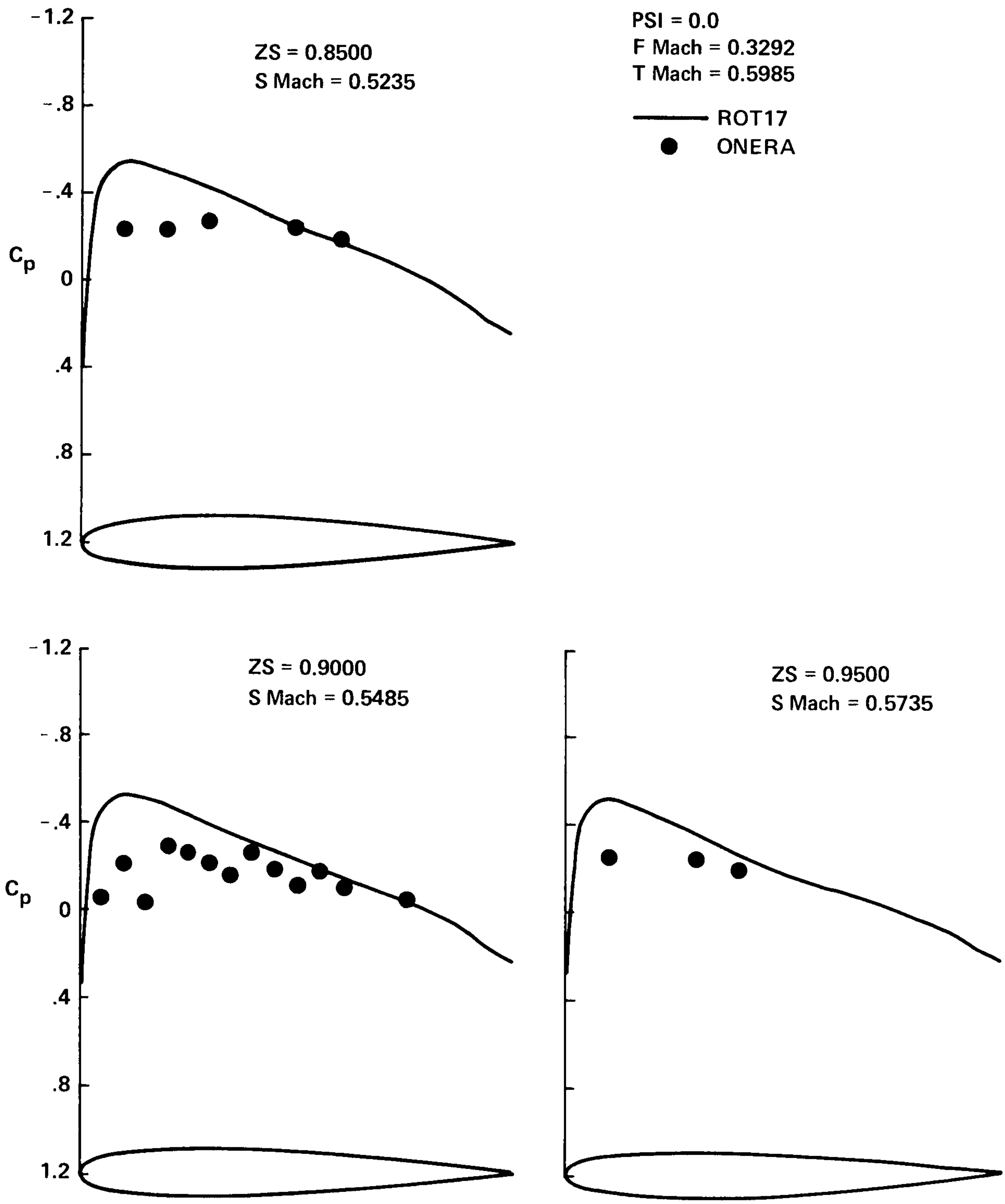
(f) Advance ratio 0.4 at 150° azimuthal angle.

Figure 4.- Continued.



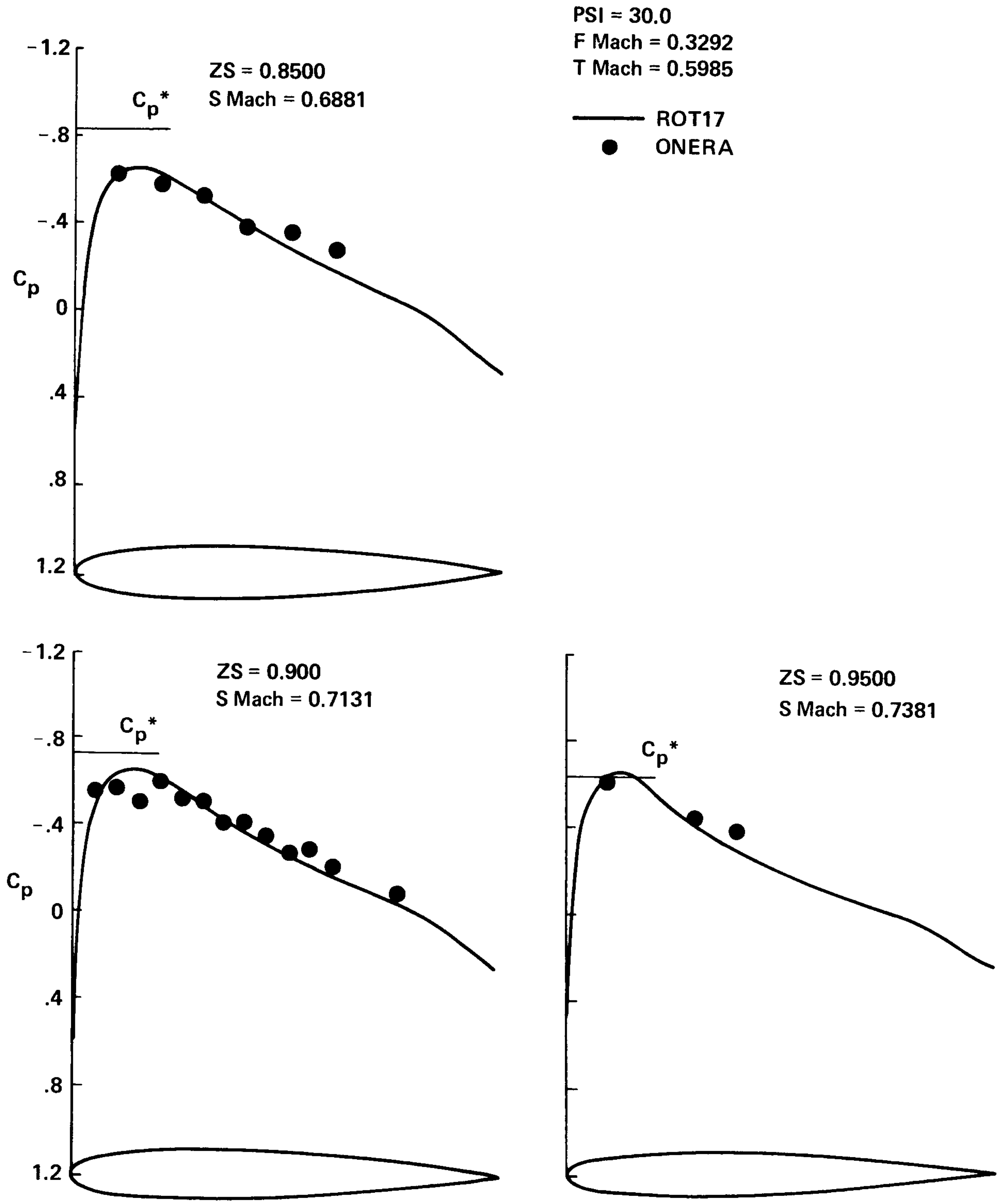
(g) Advance ratio 0.4 at 180° azimuthal angle.

Figure 4.- Concluded.



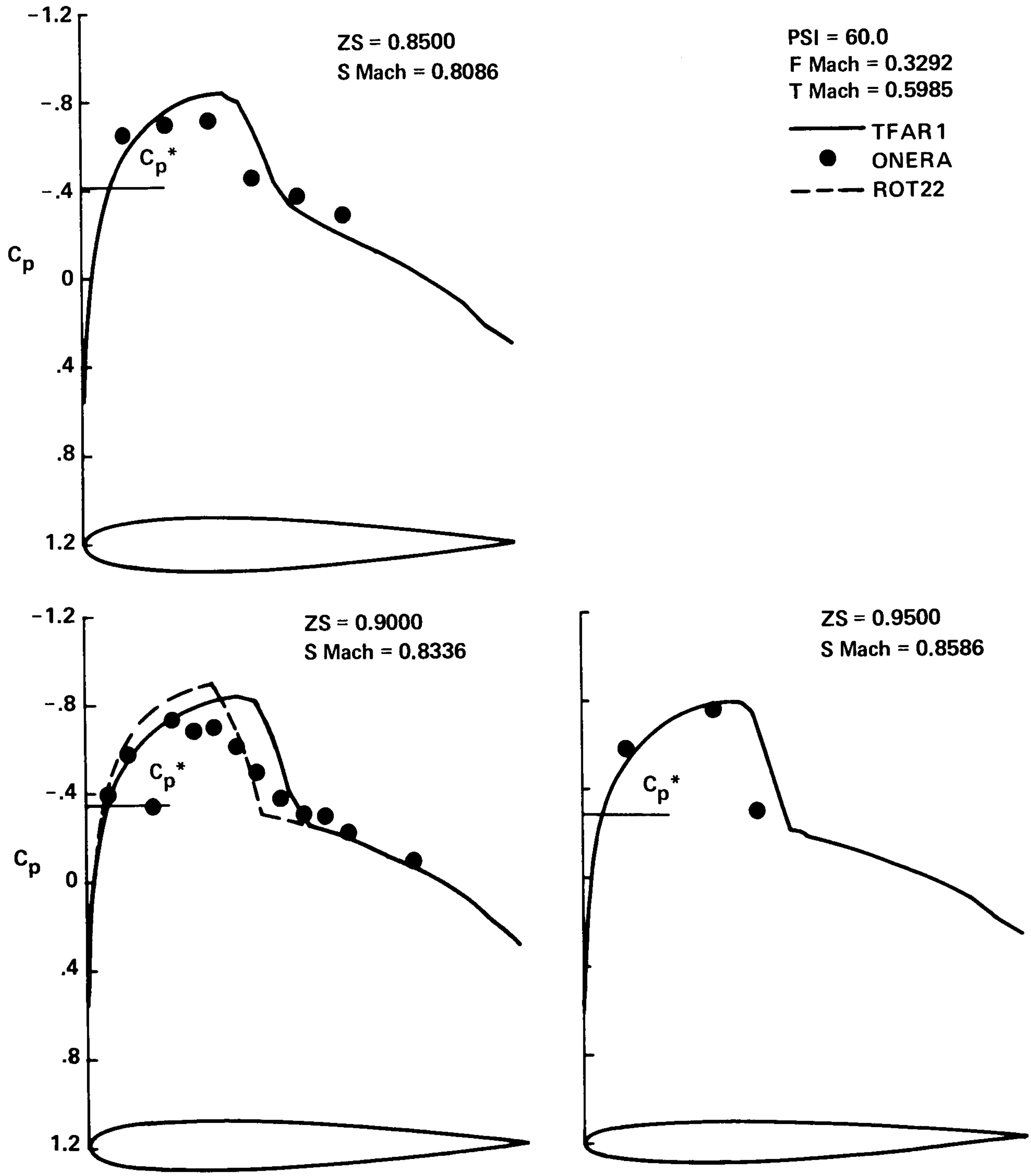
(a) Advance ratio 0.55 at 0° azimuthal angle.

Figure 5.- Comparison between computed and measured surface pressure distributions.



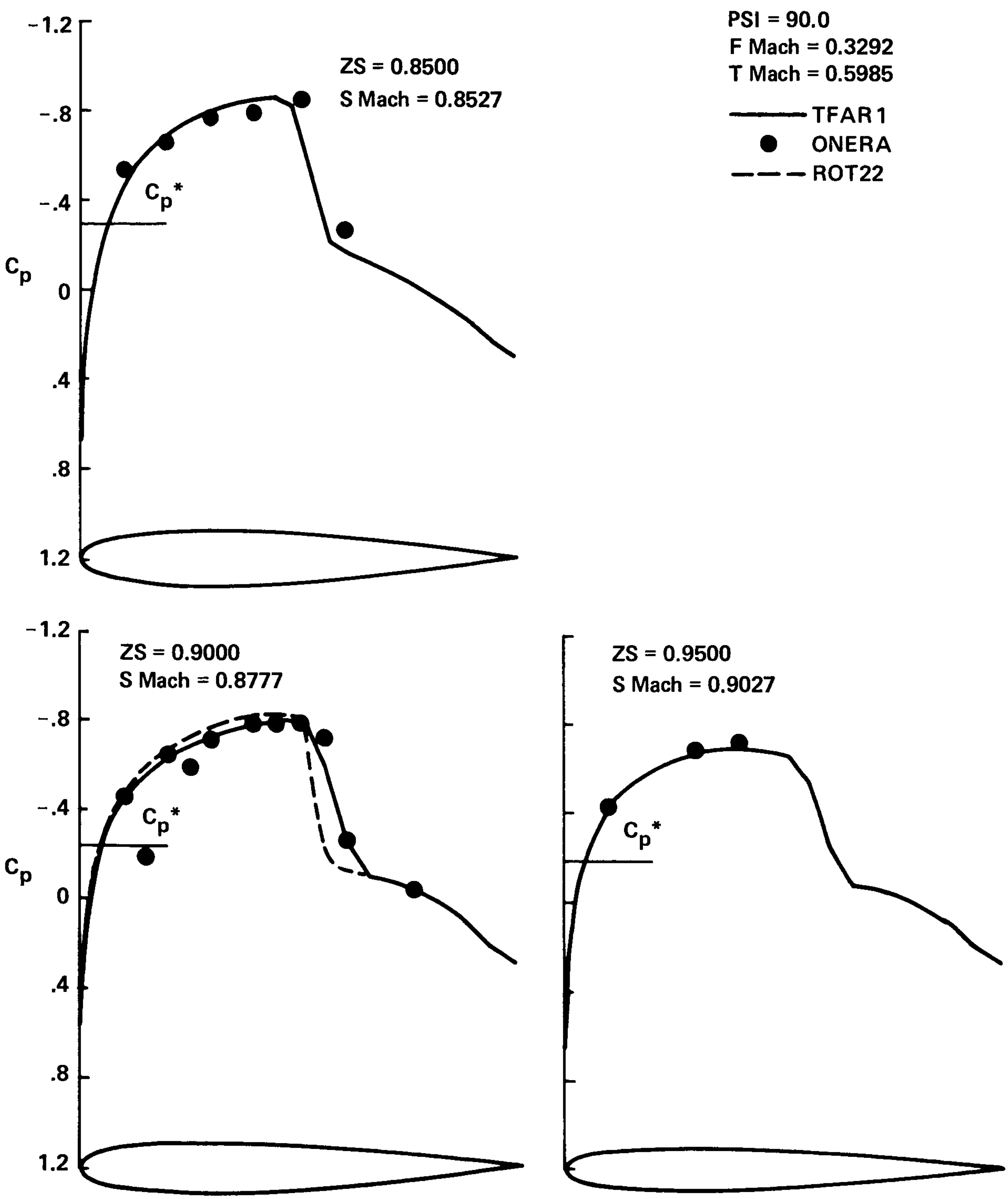
(b) Advance ratio 0.55 at 30° azimuthal angle.

Figure 5.- Continued.



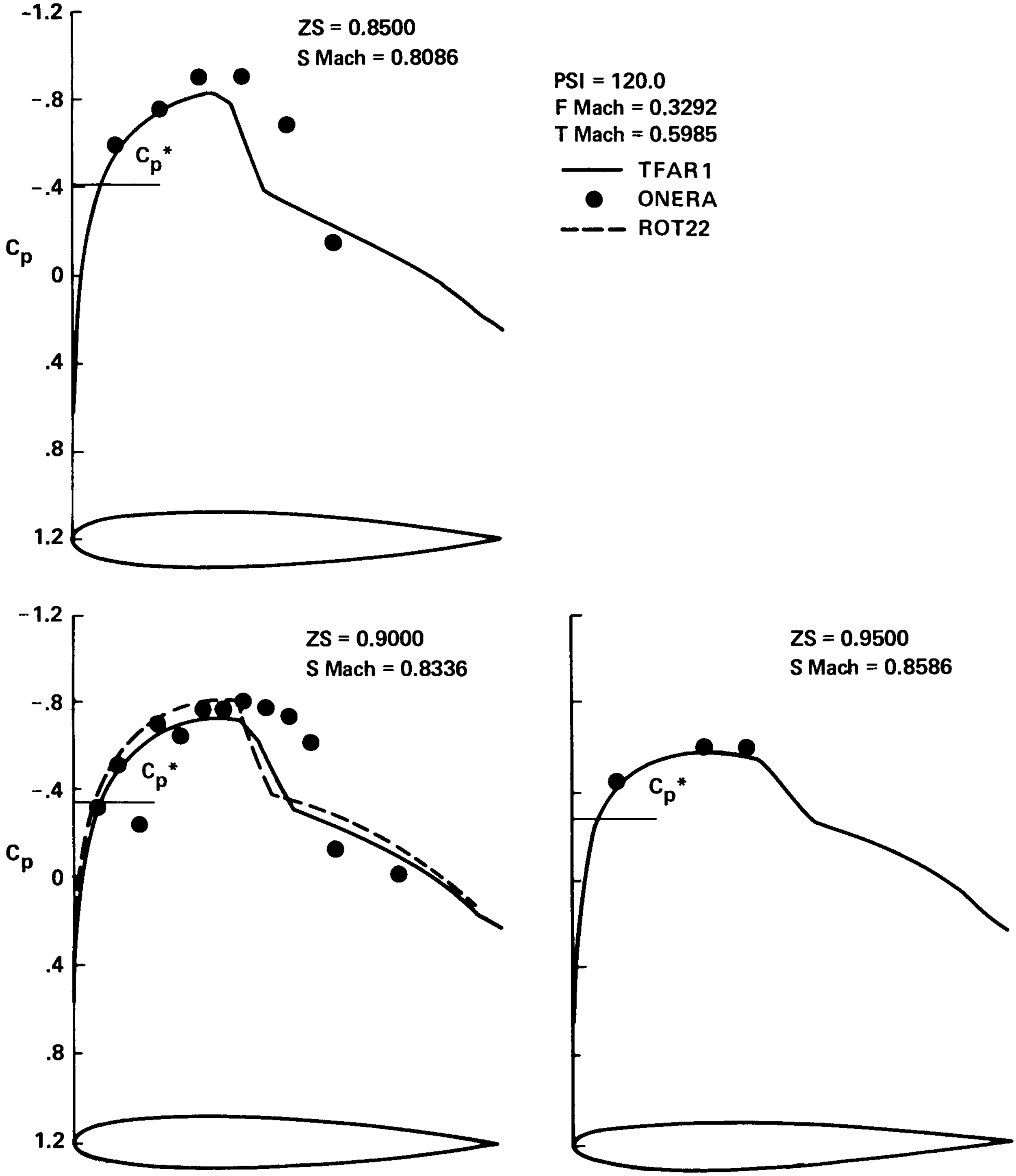
(c) Advance ratio 0.55 at 60° azimuthal angle.

Figure 5.- Continued.



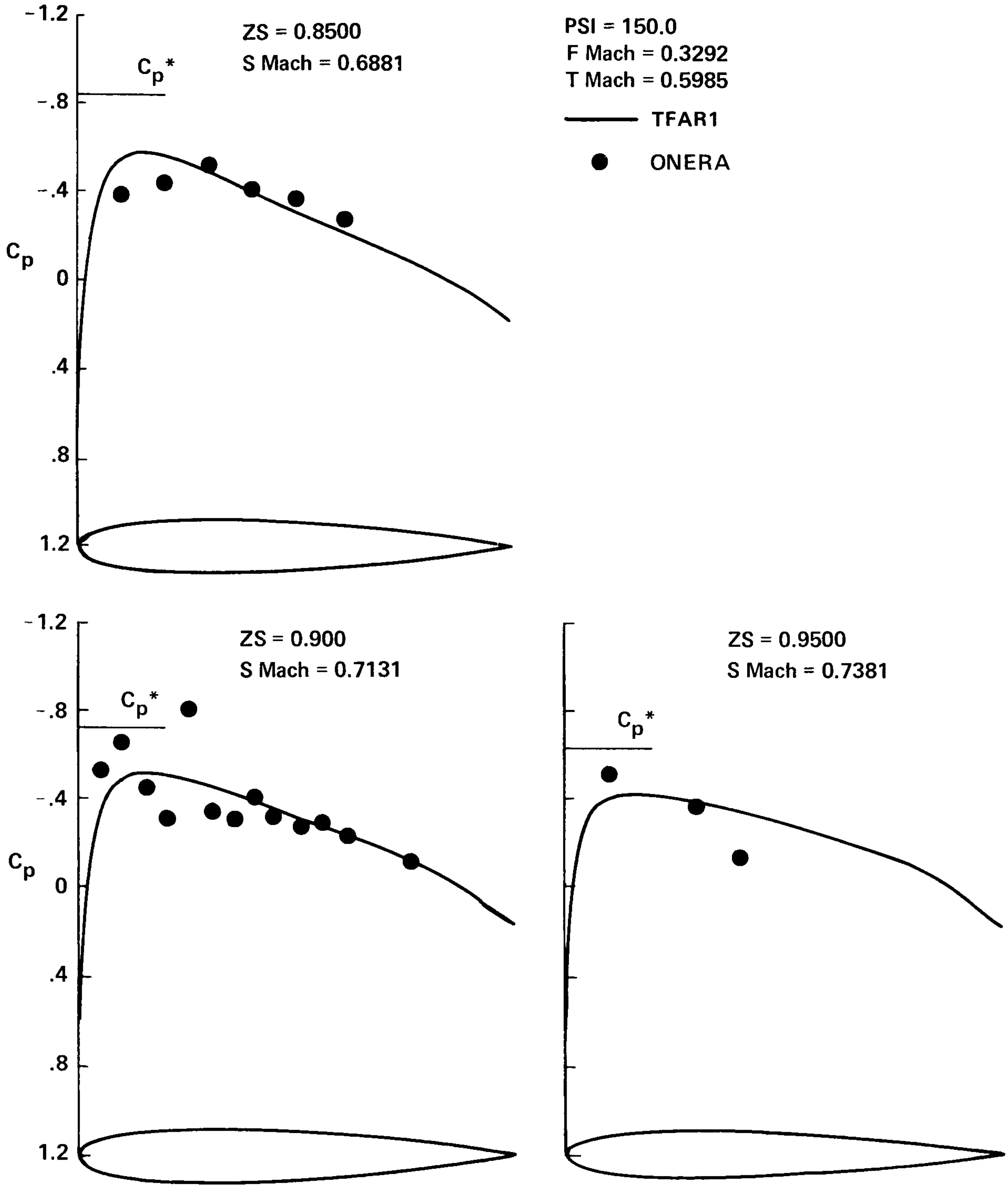
(d) Advance ratio 0.55 at 90° azimuthal angle.

Figure 5.- Continued.



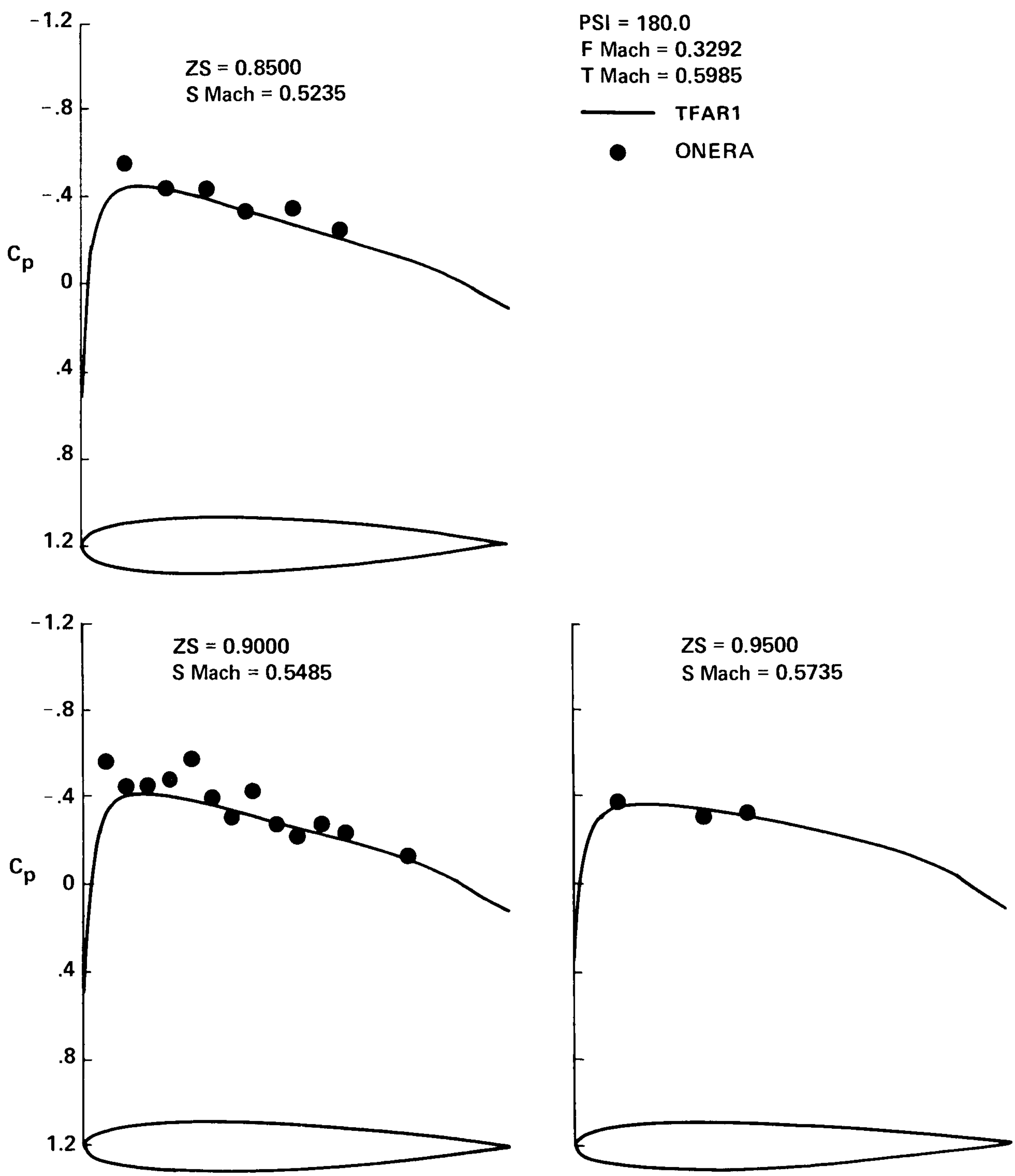
(e) Advance ratio 0.55 at 120° azimuthal angle.

Figure 5.- Continued.



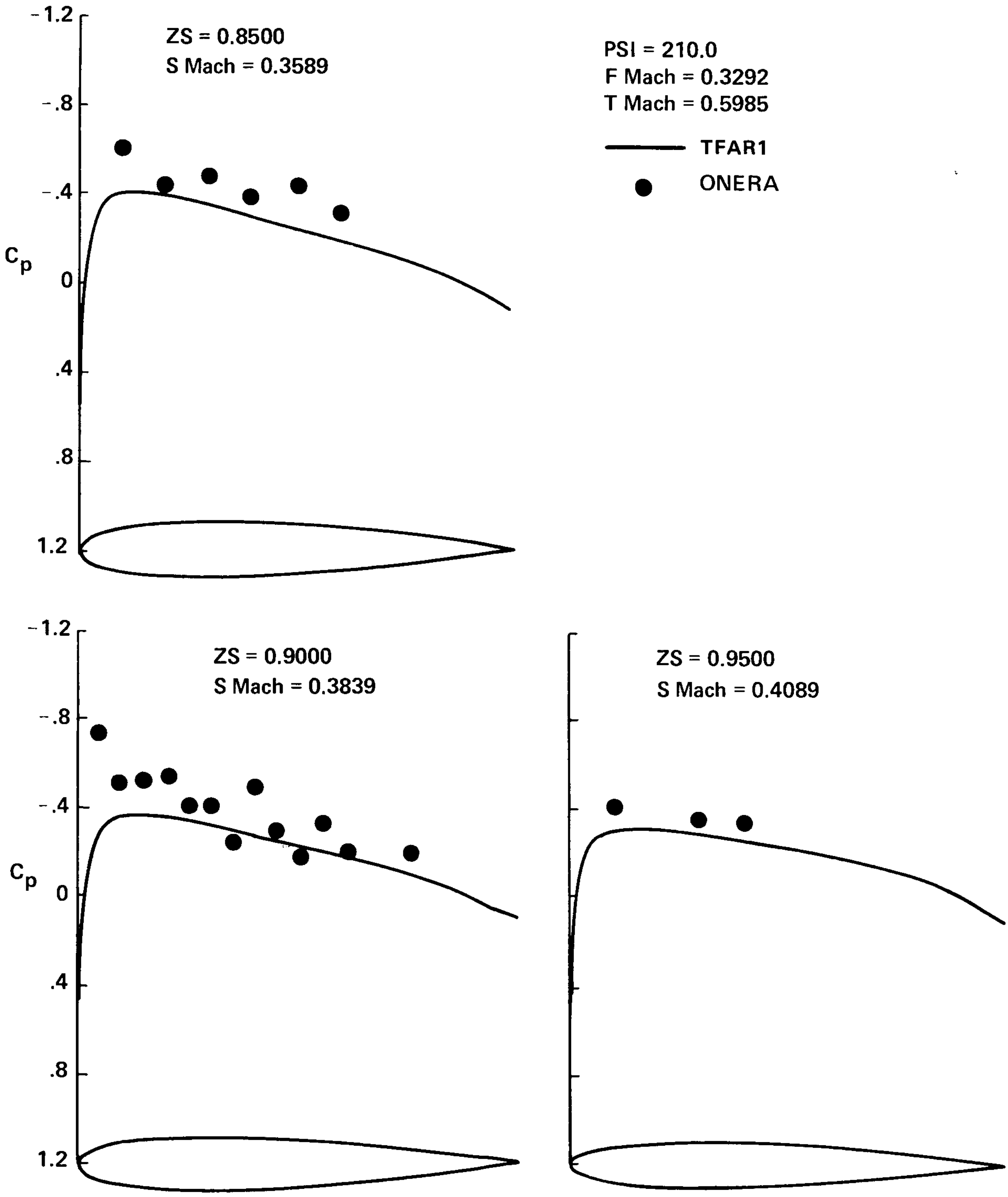
(f) Advance ratio 0.55 at 150° azimuthal angle.

Figure 5.- Continued.



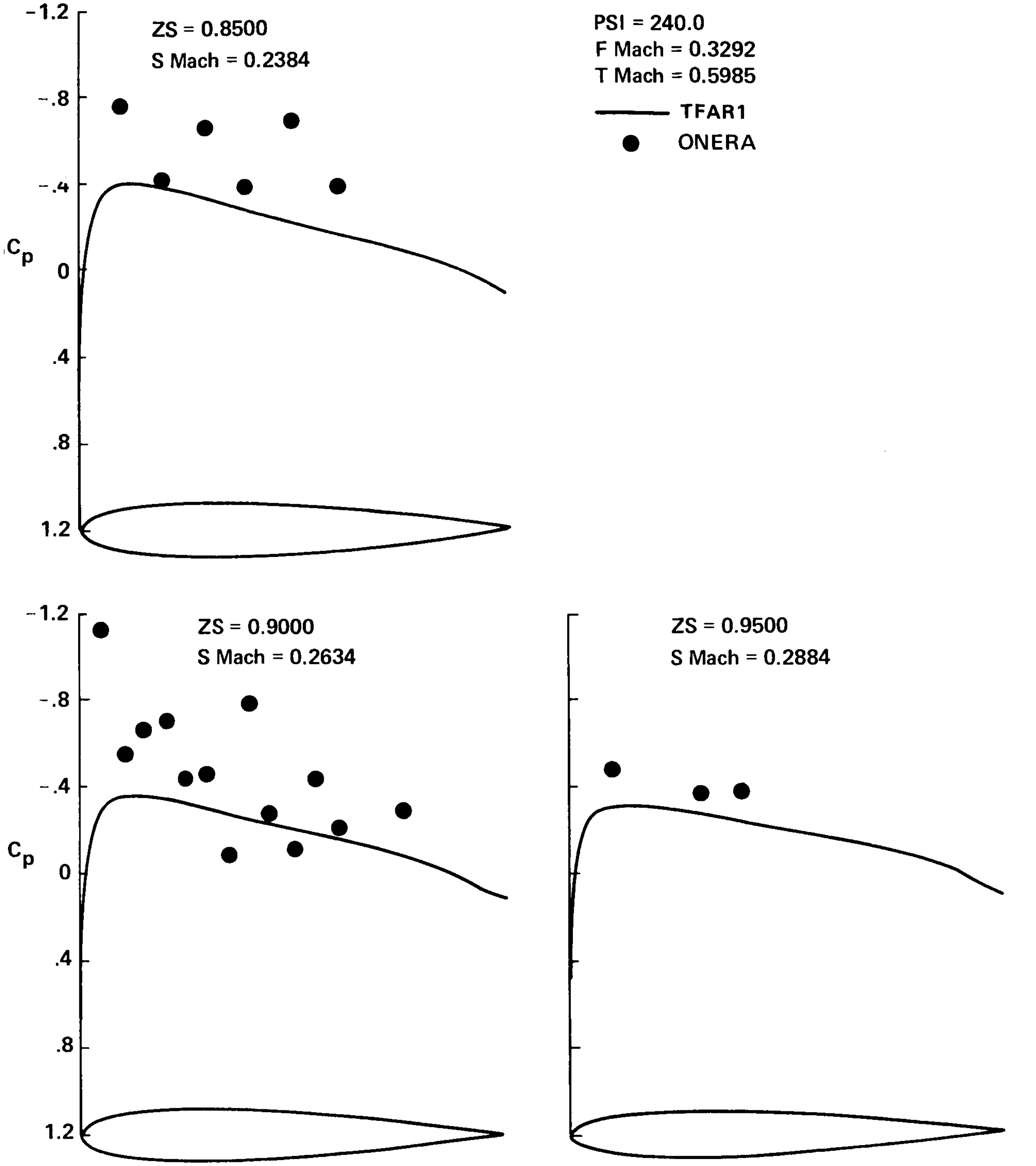
(g) Advance ratio 0.55 at 180° azimuthal angle.

Figure 5.- Continued.



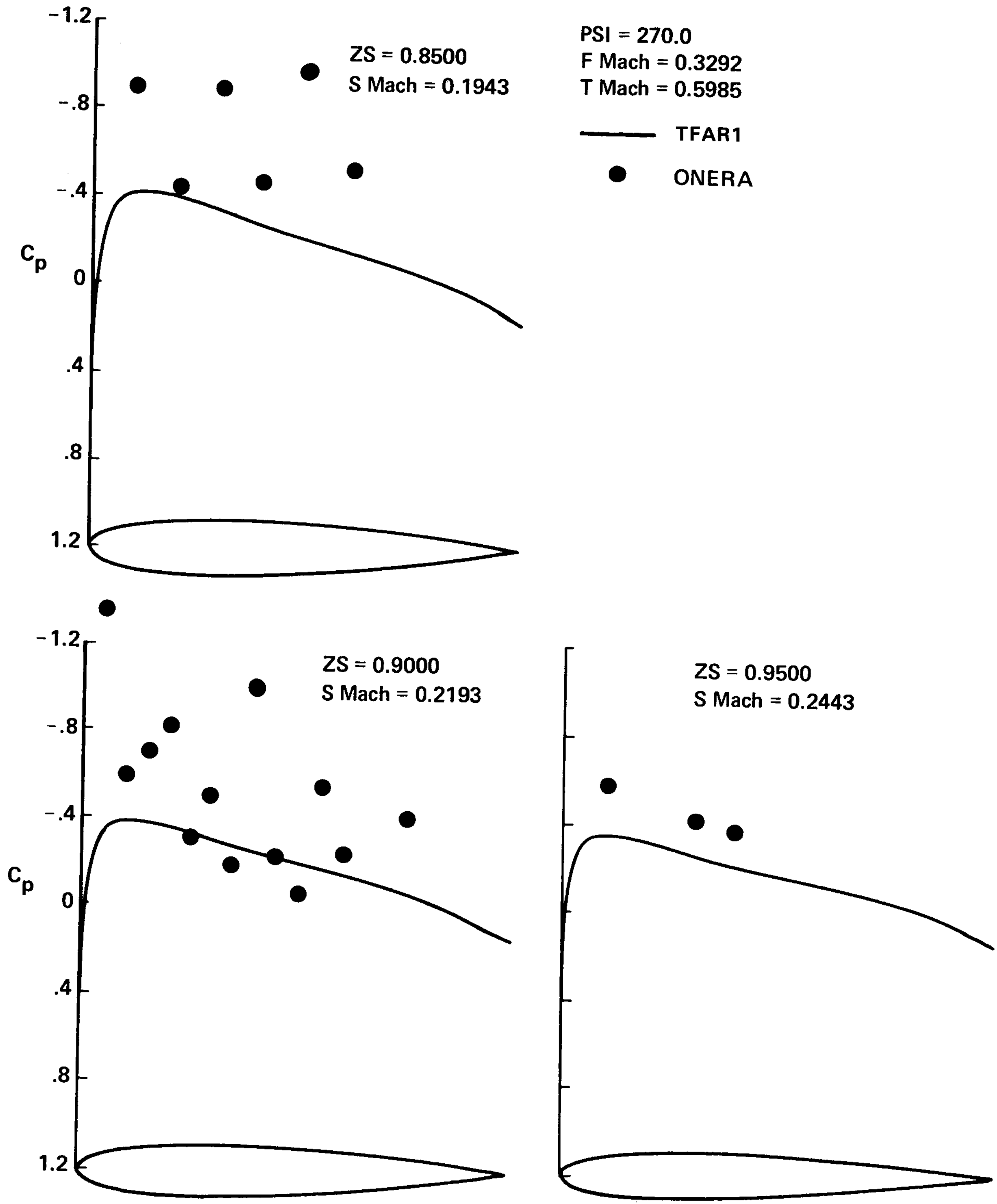
(h) Advance ratio 0.55 at 210° azimuthal angle.

Figure 5.- Continued.



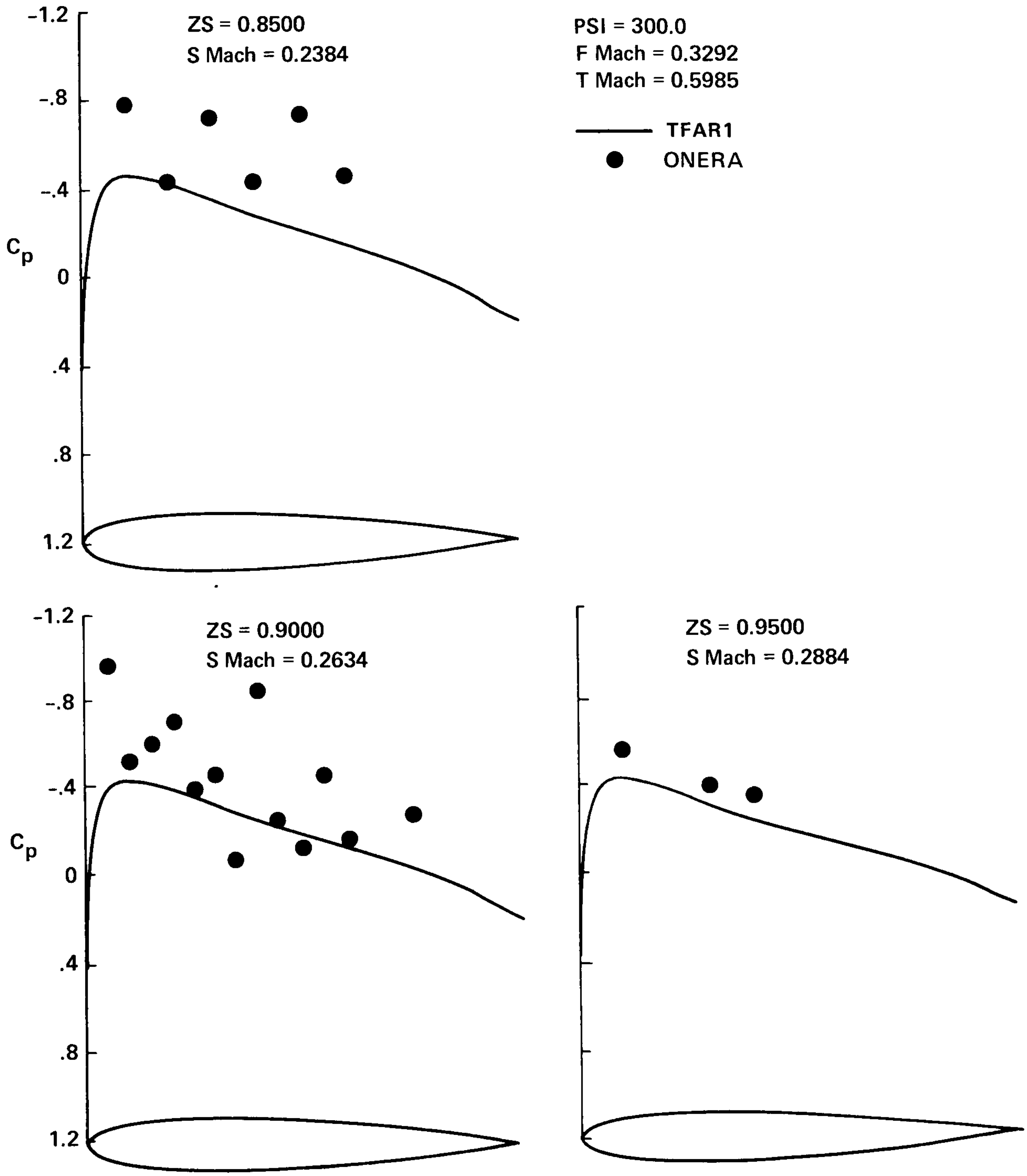
(i) Advance ratio 0.55 at 240° azimuthal angle.

Figure 5.- Continued.



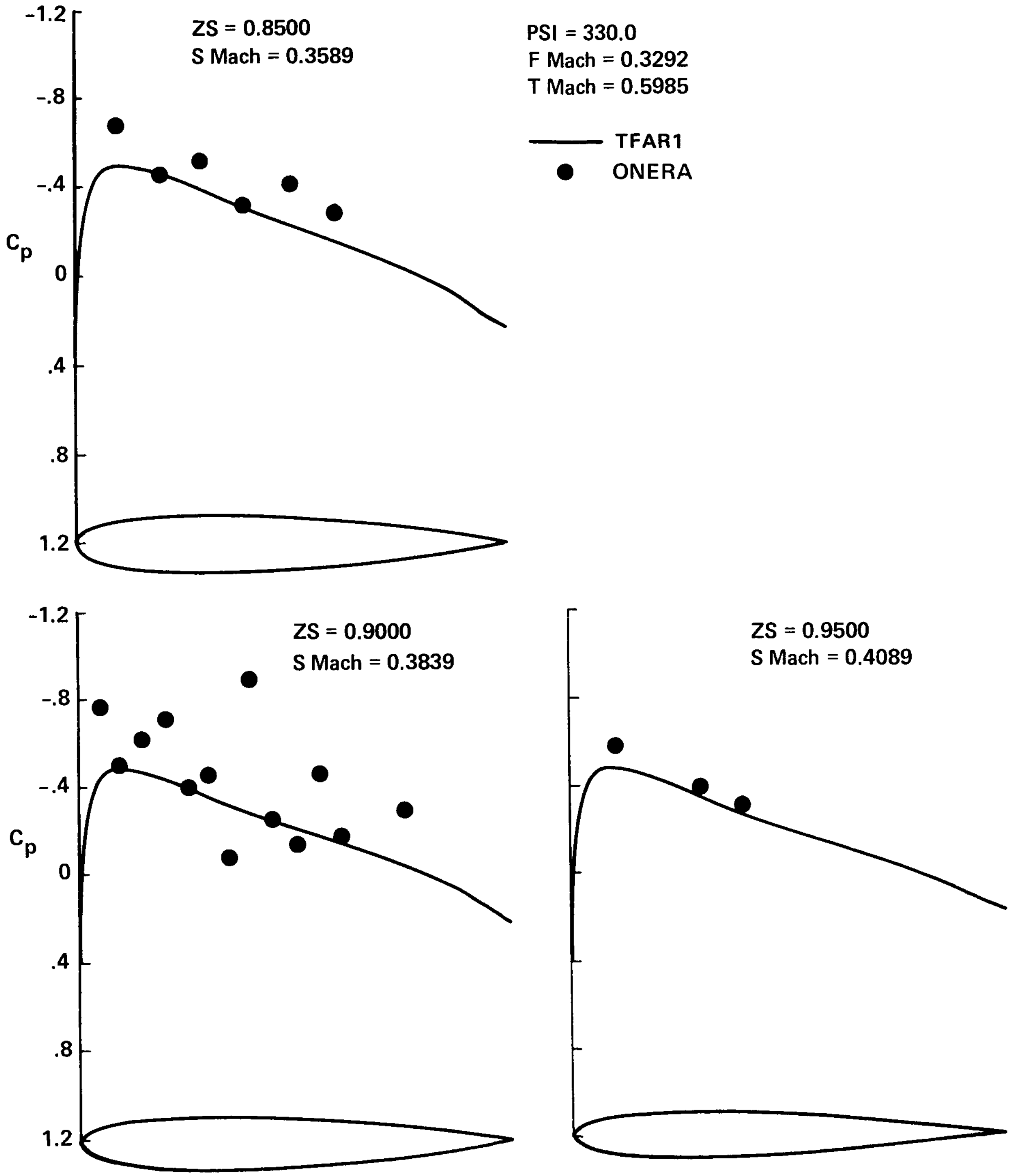
(j) Advance ratio 0.55 at 270° azimuthal angle.

Figure 5.- Continued.



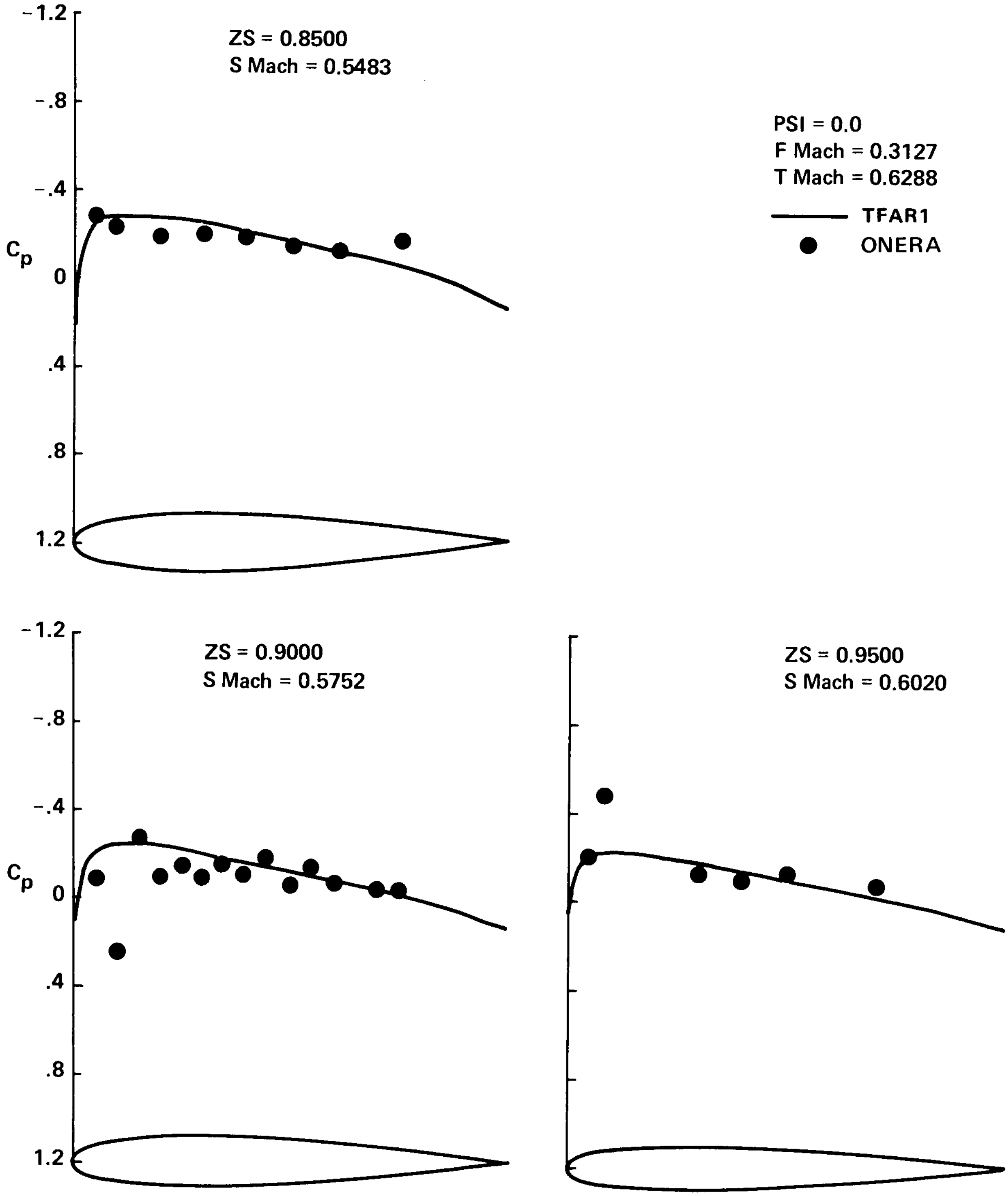
(k) Advance ratio 0.55 at 300° azimuthal angle.

Figure 5.- Continued.



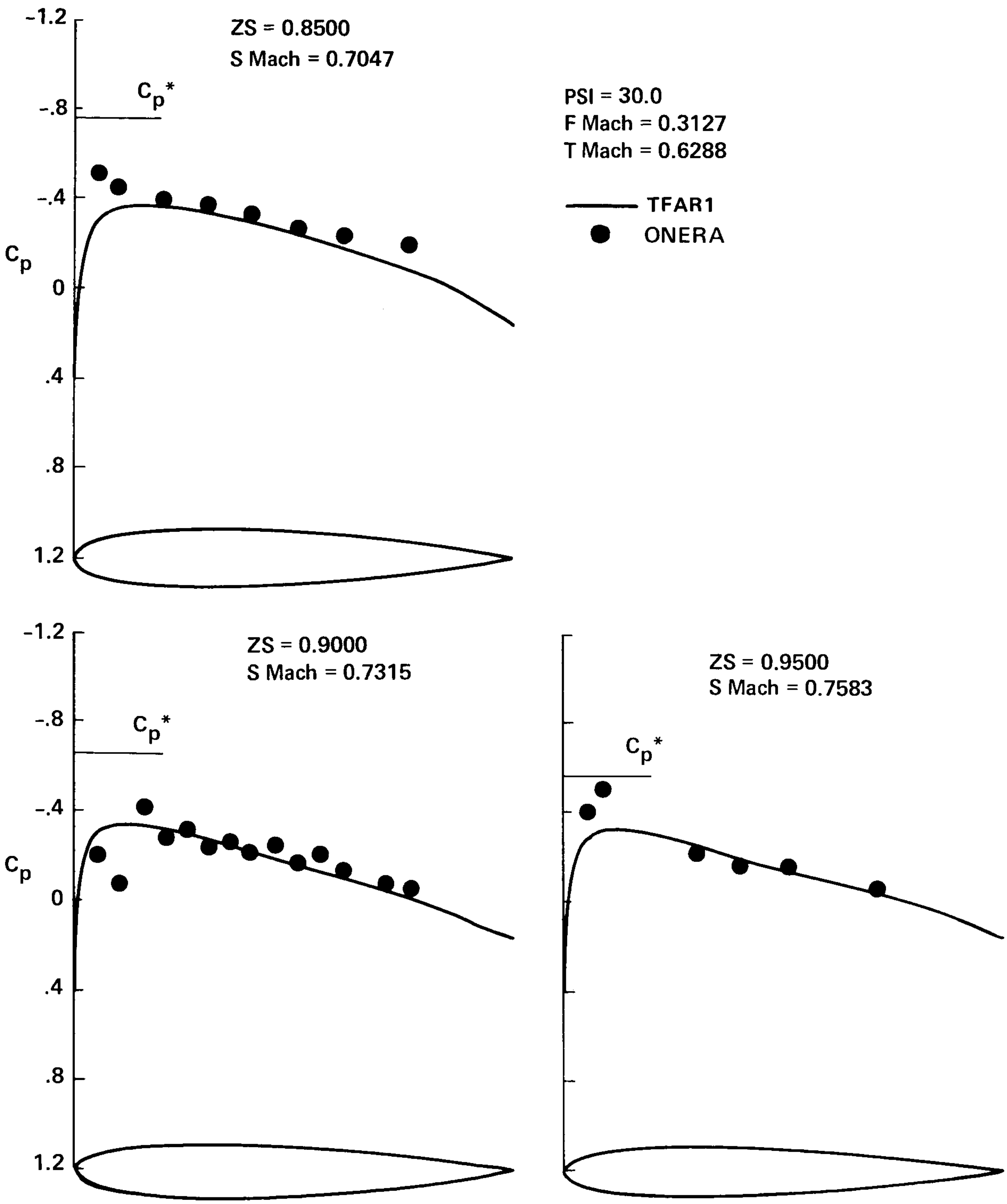
(1) Advance ratio 0.55 at 330° azimuthal angle.

Figure 5.- Concluded.



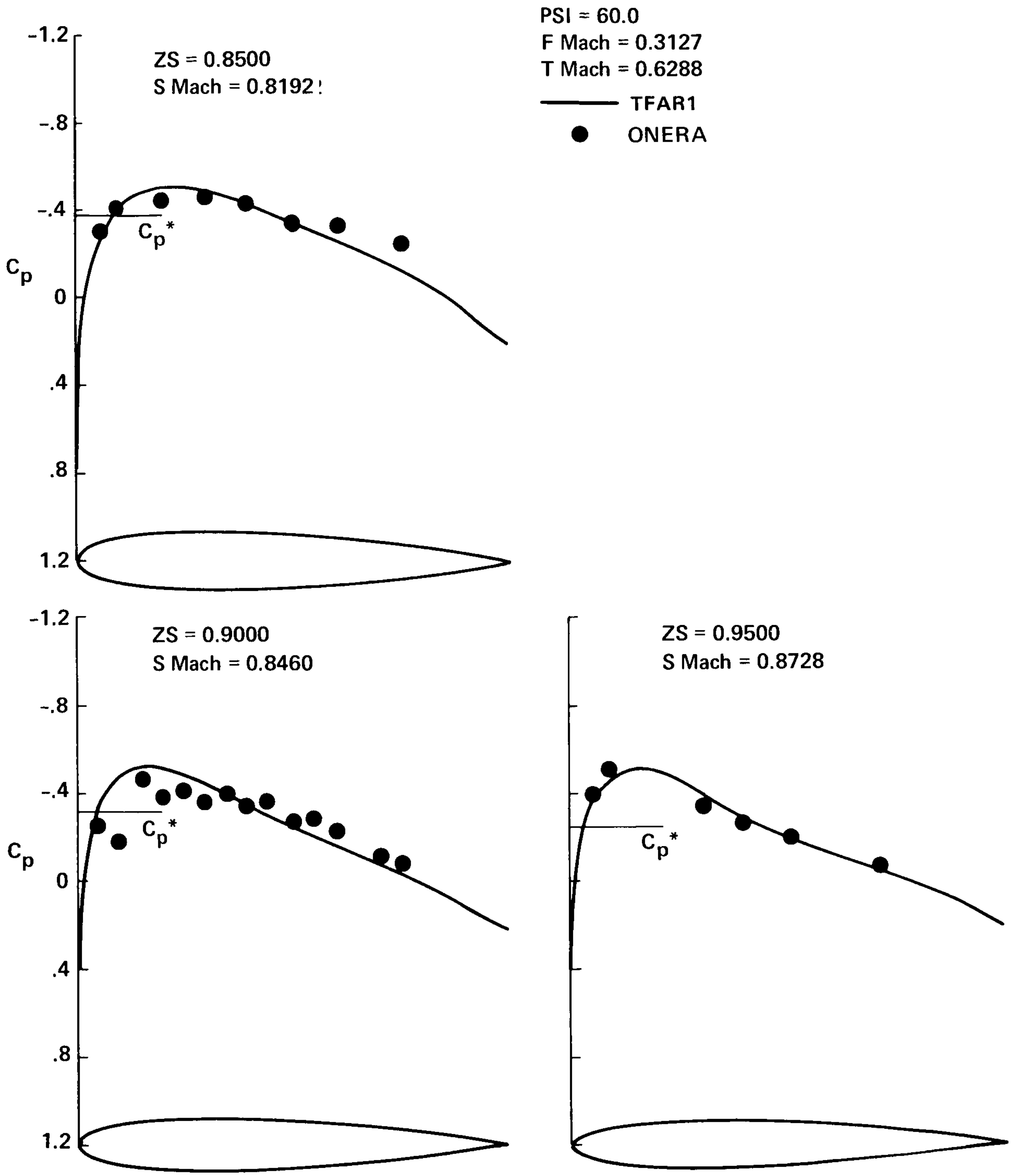
(a) Advance ratio 0.5 at 0° azimuthal angle.

Figure 6.- Comparison between computed and measured surface pressure distributions.



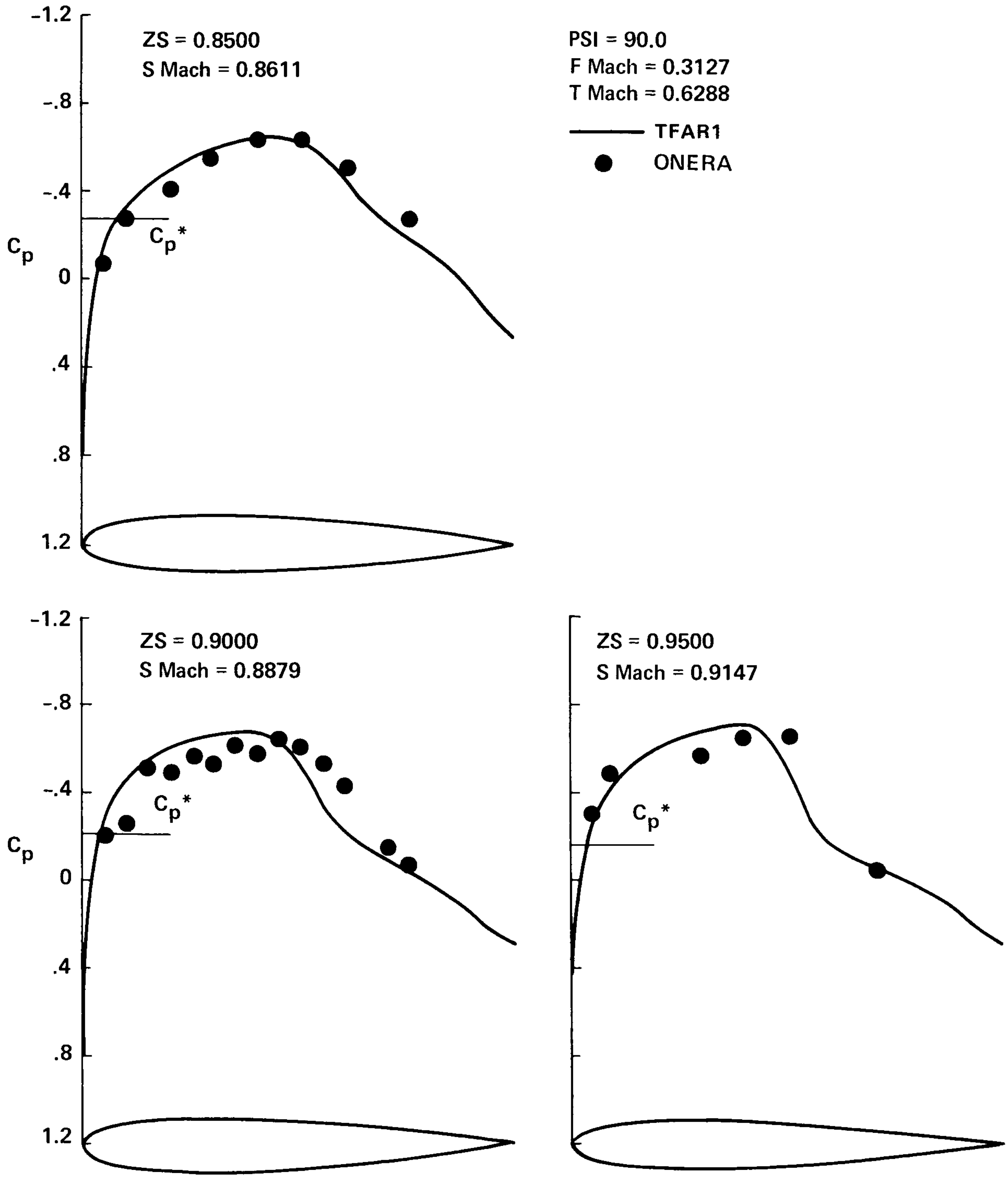
(b) Advance ratio 0.5 at 30° azimuthal angle.

Figure 6.- Continued.



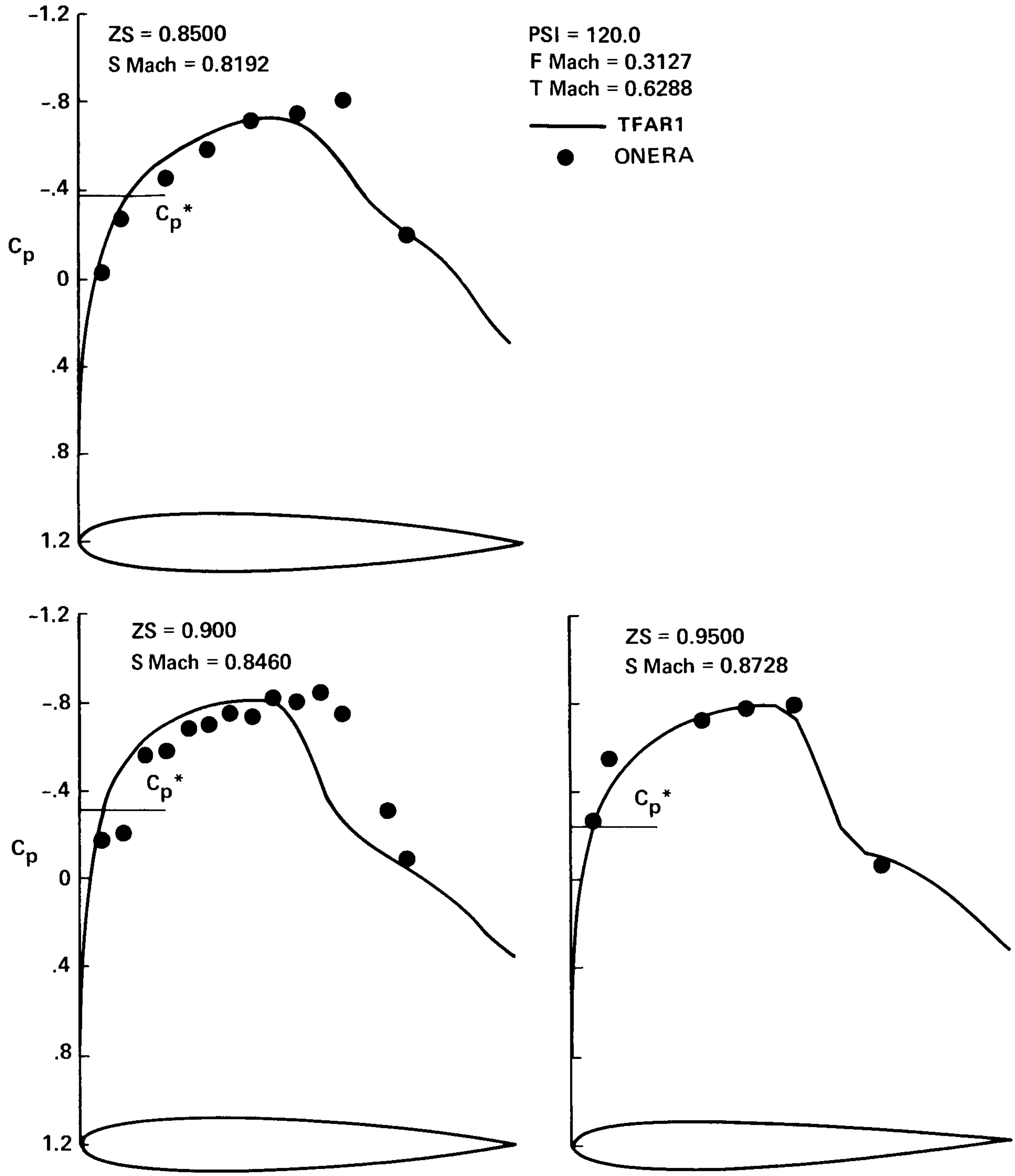
(c) Advance ratio 0.5 at 60° azimuthal angle.

Figure 6.- Continued.



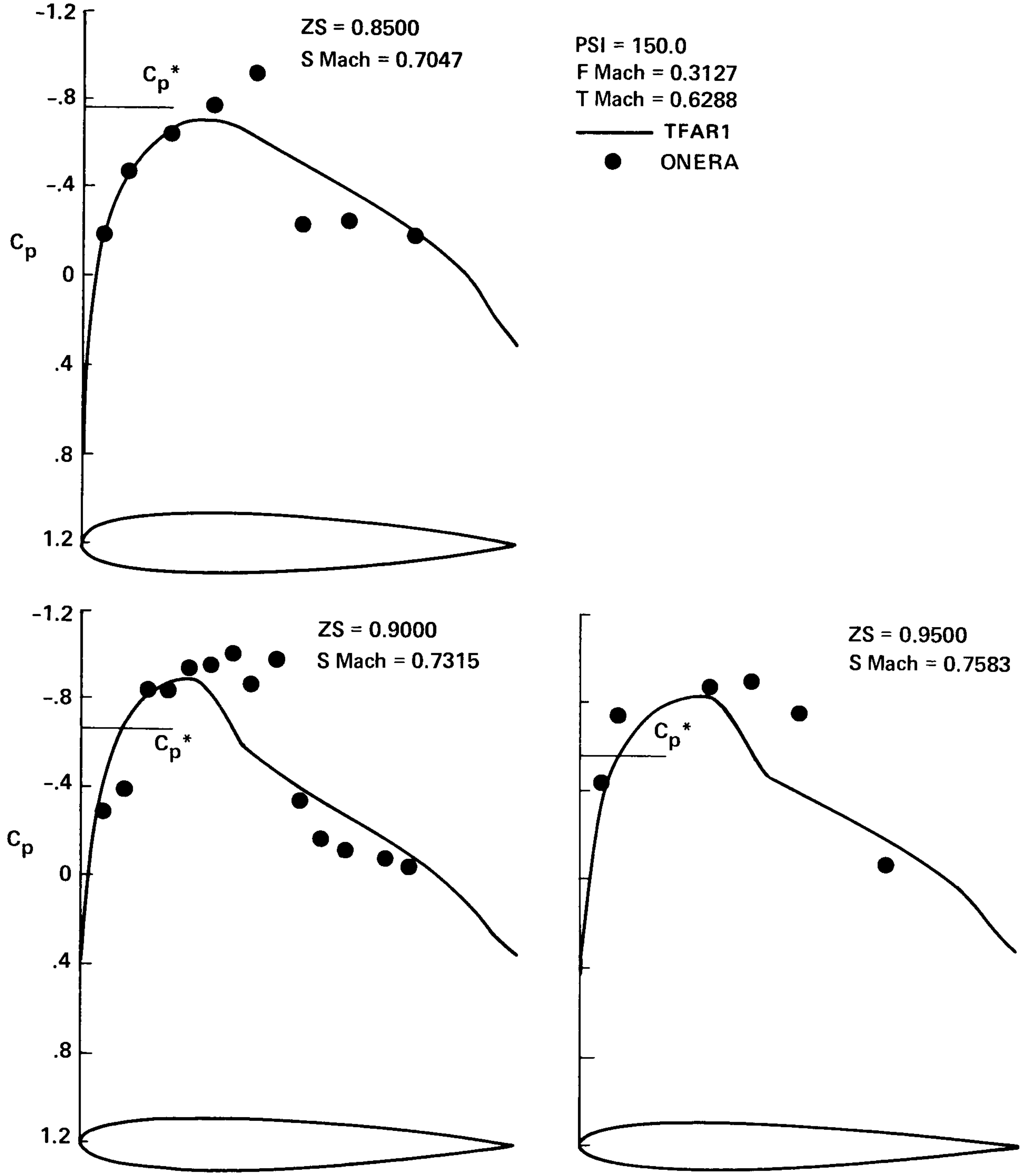
(d) Advance ratio 0.5 at 90° azimuthal angle.

Figure 6.- Continued.



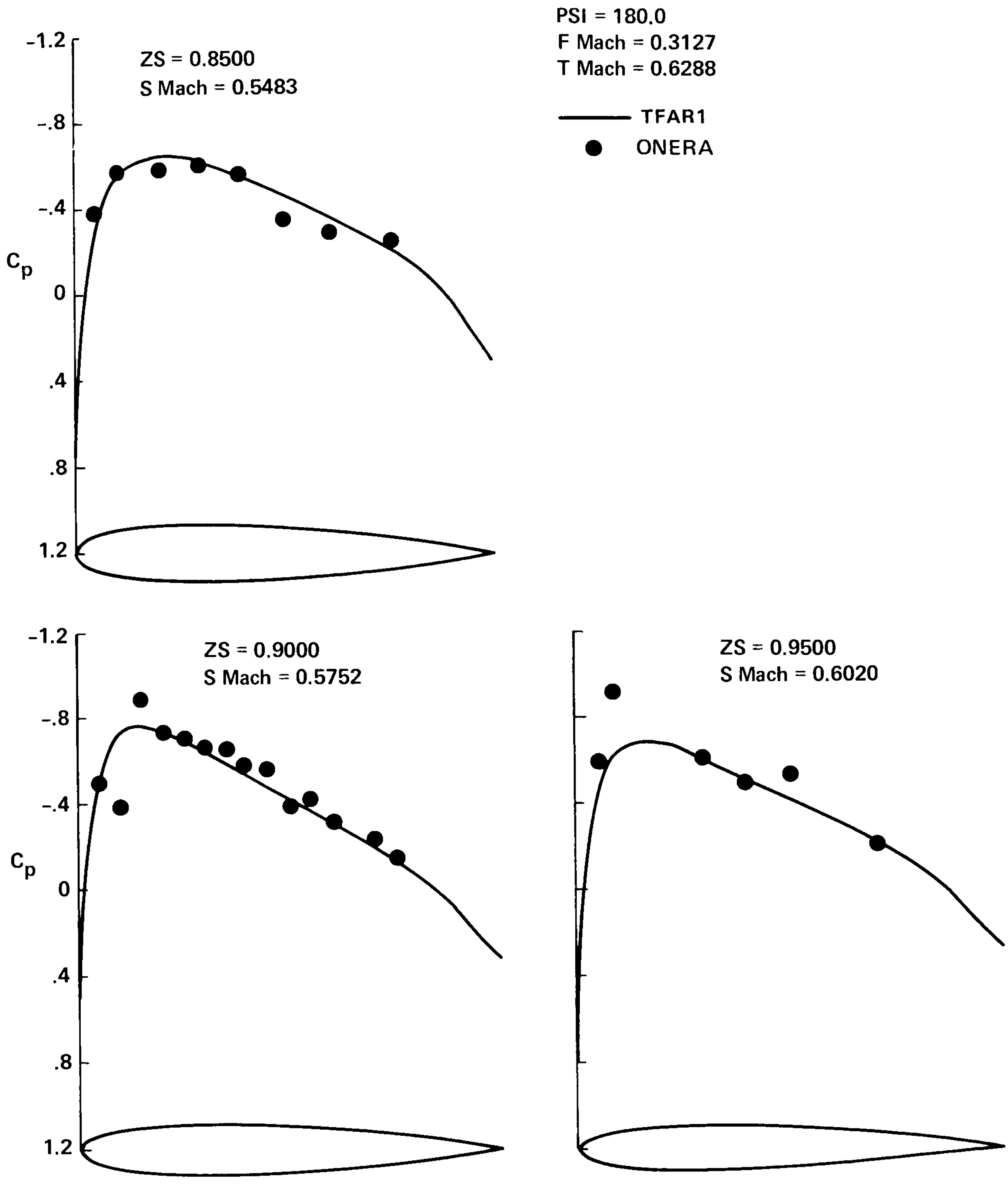
(e) Advance ratio 0.5 at 120° azimuthal angle.

Figure 6.- Continued.



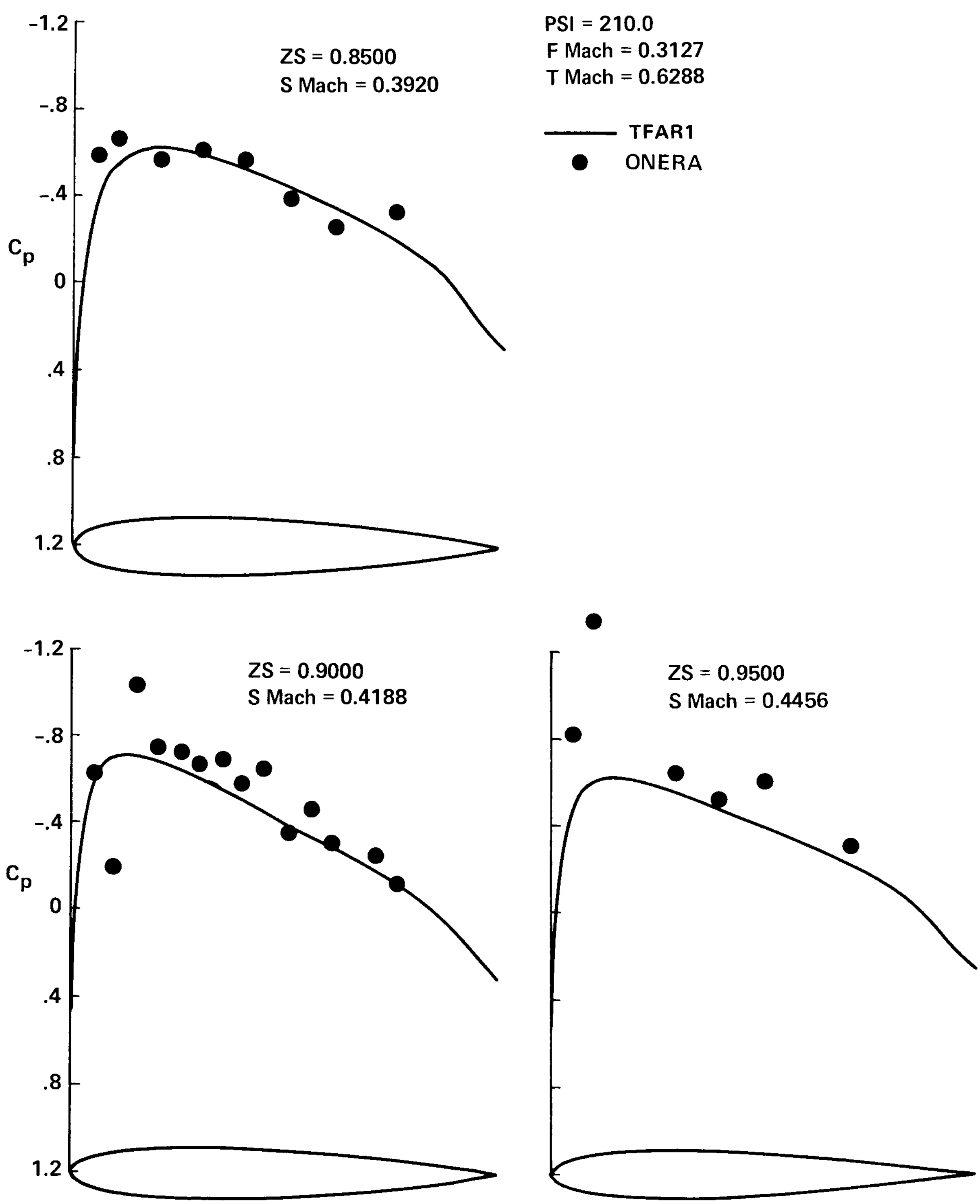
(f) Advance ratio 0.5 at 150° azimuthal angle.

Figure 6.- Continued.



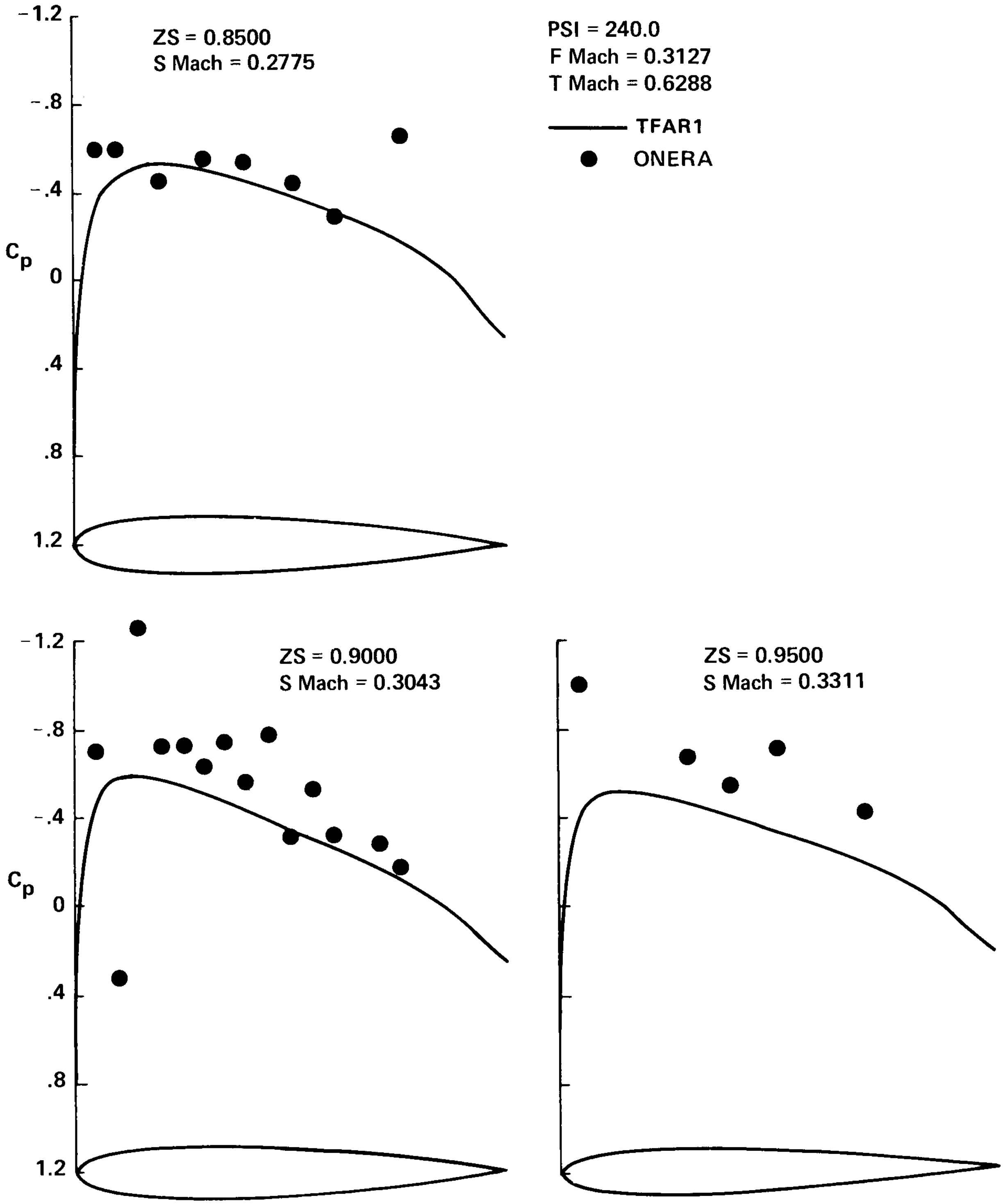
(g) Advance ratio 0.5 at 180° azimuthal angle.

Figure 6.- Continued.



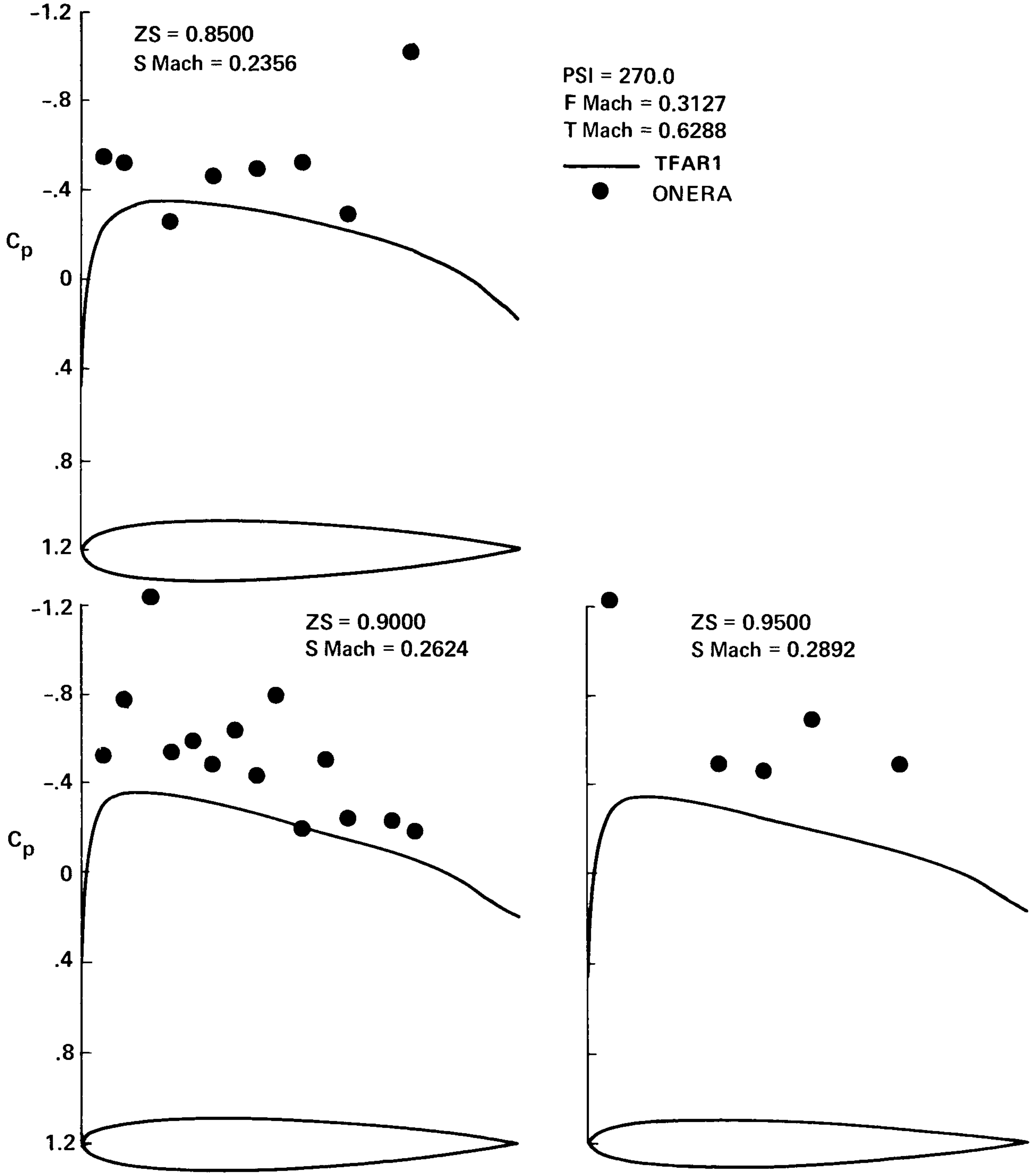
(h) Advance ratio 0.5 at 210° azimuthal angle.

Figure 6.- Continued.



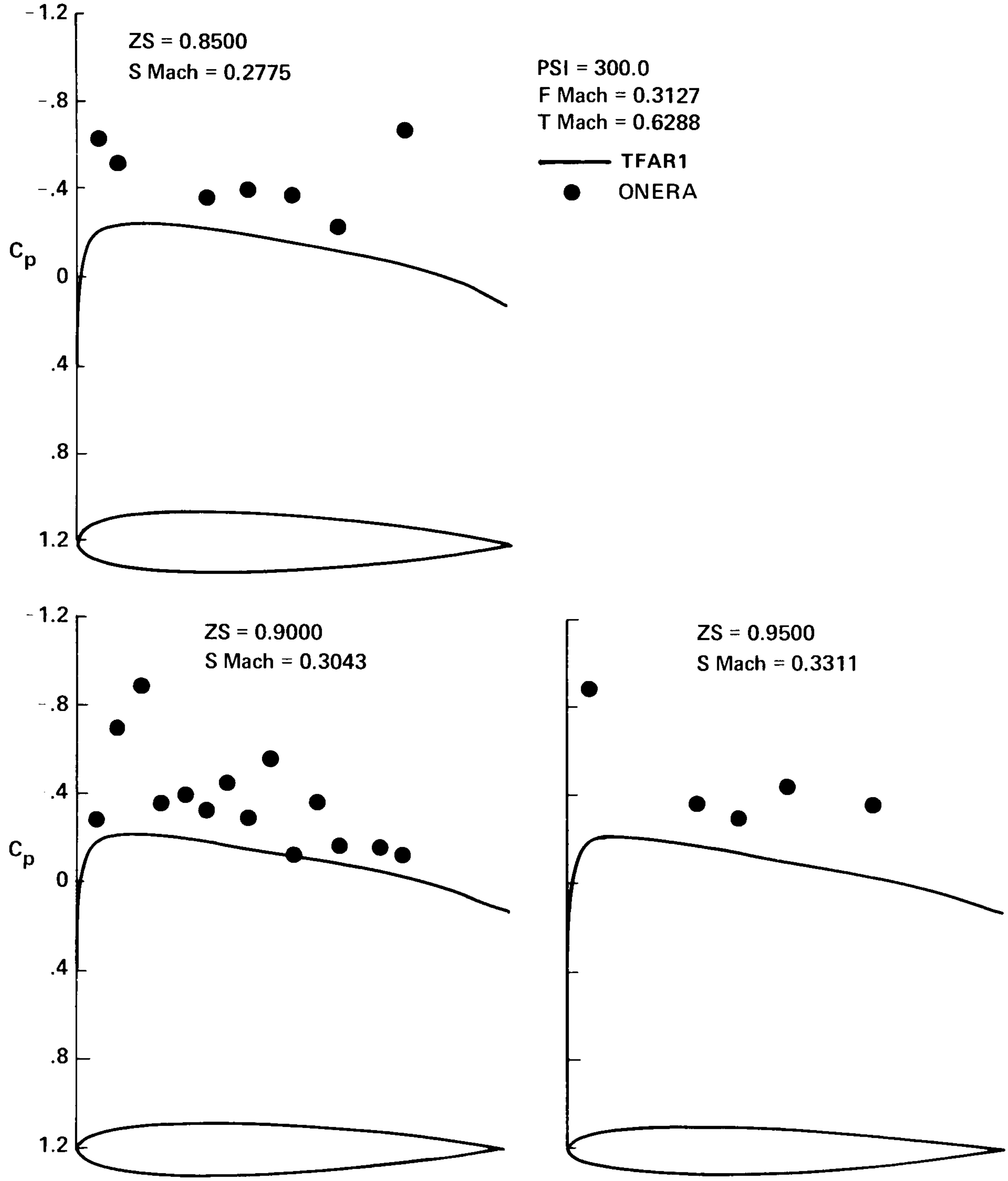
(i) Advance ratio 0.5 at 240° azimuthal angle.

Figure 6.- Continued.



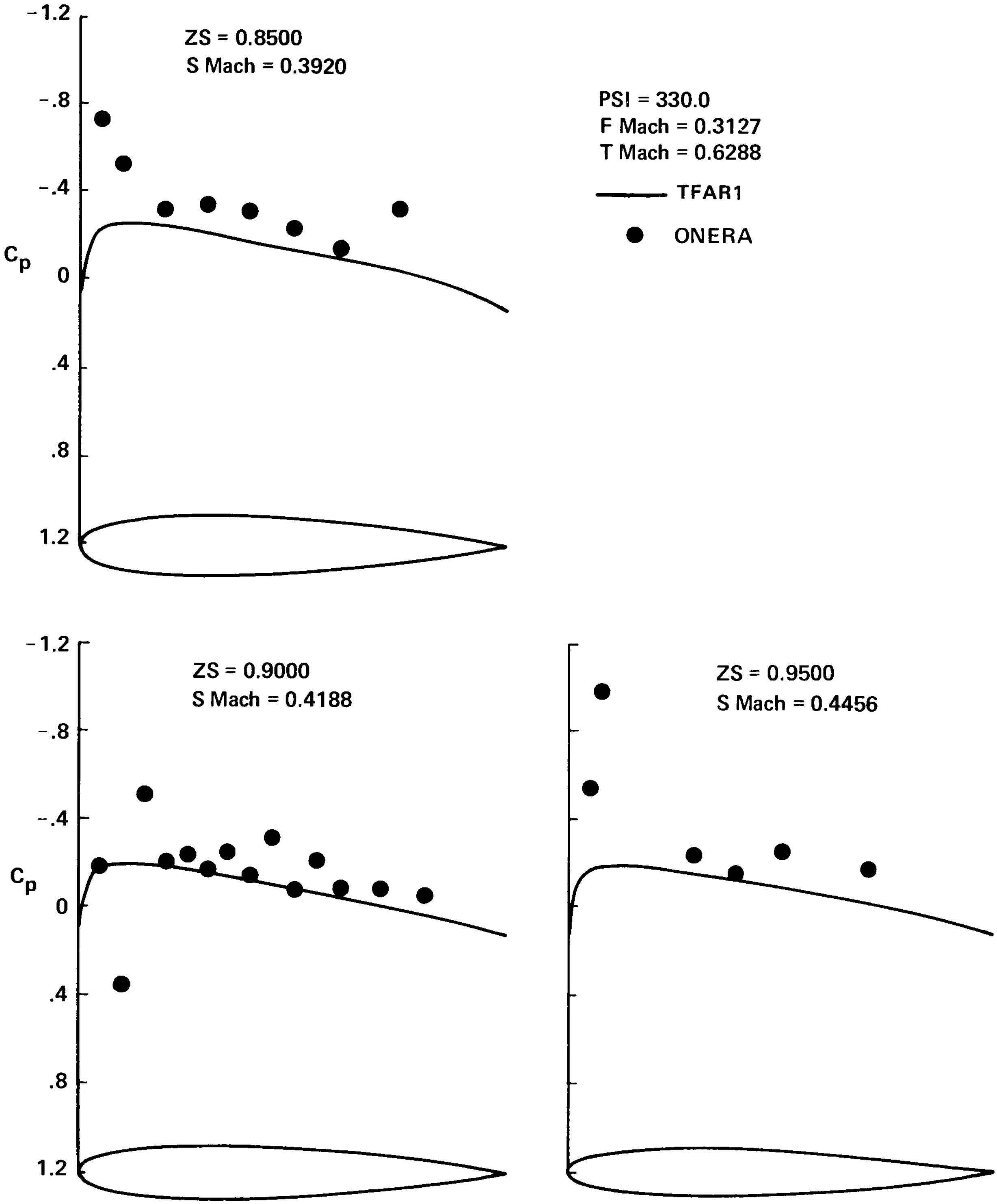
(j) Advance ratio 0.5 at 270° azimuthal angle.

Figure 6.- Continued.



(k) Advance ratio 0.5 at 300° azimuthal angle.

Figure 6.- Continued.



(l) Advance ratio 0.5 at 330° azimuthal angle.

Figure 6.- Concluded.

1. Report No. NASA TP-2375	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle TRANSONIC FLOW ANALYSIS FOR ROTORS. PART I — THREE-DIMENSIONAL, QUASI-STEADY, FULL-POTENTIAL CALCULATION		5. Report Date October 1984
		6. Performing Organization Code
7. Author(s) I-Chung Chang		8. Performing Organization Report No. A-9721
9. Performing Organization Name and Address NASA Ames Research Center Moffett Field, CA 94035		10. Work Unit No. T-3414Y
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Technical Paper
		14. Sponsoring Agency Code 505-42-11

15. Supplementary Notes

Point of Contact: I-Chung Chang, Ames Research Center, MS 227-8, Moffett Field, CA 94035. (415) 965-6396 or FTS 448-6396.

16. Abstract

A new computer program is presented for calculating the quasi-steady transonic flow past a helicopter rotor blade in hover as well as in forward flight. The program is based on the full potential equations in a blade-attached frame of reference and is capable of treating a very general class of rotor blade geometries. Computed results show good agreement with available experimental data for both straight- and swept-tip blade geometries.

17. Key Words (Suggested by Author(s)) Aerodynamics Transonic flow Helicopter aerodynamics	18. Distribution Statement Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 86	22. Price A03

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1984

National Aeronautics and Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

2 2 1U,A, 841012 S00161DS
DEPT OF THE AIR FORCE
ARNOLD ENG DEVELOPMENT CENTER(AFSC)
ATTN: LIBRARY/DOCUMENTS
ARNOLD AF STA TN 37389

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return